COMPASS: Prefix Realization under Routing Preferences

A step-by-step account of the conflict graph and its bounds

Loqman Salamatian

November 2026

Abstract

COMPASS selects ingress infrastructure and steers traffic using a limited number of BGP prefixes. Unlike direct traffic assignment, a prefix exposes the same set of announcing ingresses to every user group using it; each group then chooses its preferred ingress. We formalize the deployment and routing decisions, then study a two-stage approach: first compute a desirable assignment assuming direct control, and then determine how many prefixes are needed to reproduce it. A conflict graph captures exactly which intended assignments cannot share a prefix, making the minimum number of prefixes its chromatic number. We show that even second-choice assignments can require arbitrarily many prefixes, although never more than logarithmically many in the number of ingresses used, and identify two kinds of preference agreement that restore constant bounds. Finally, we address the case where the available prefix budget is too small to preserve every assignment.

1 A prefix is a shared menu

Suppose COMPASS would like to send user group U through ingress B, although U ranks the options $A \succ_U B \succ_U C$. If only B announces the prefix given to U, then the resulting route is B. If both A and B announce it, the resulting route is A. COMPASS chooses the *set of visible options* and then users select which of these options they pick (hence the menu analogy!).

This distinction matters when prefixes are shared. Putting two user groups on the same prefix makes both groups see the union of the ingresses needed by their intended assignments. An ingress desired by one group may be preferred over the intended ingress of the other. In that case, the two assignments cannot both be preserved by that prefix.

There are therefore two separate questions:

1. **Infrastructure selection:** which facilities and ingress options should carry the traffic if COMPASS could assign demand directly?
2. **Prefix realization:** with those options selected, which user assignments can share a BGP prefix without changing the resulting routes?

We first state the joint model to make its assumptions explicit. Figure 1 then explains how its selected flow becomes the assignment grid used to construct a conflict graph. The main combinatorial results concern this graph for a fixed deployment and target assignment.

2 Optimization model and decomposition

2.1 Inputs and decisions

Let $\mathcal{U}$ denote user groups, with traffic weights $w_U > 0$. Let $\mathcal{P}$ be the set of peers (neighboring networks) and $\mathcal{F}$ the set of candidate facilities, with elements $f \in \mathcal{F}$. The possible ingress options are peer–facility pairs $\mathcal{O} \subseteq \mathcal{P} \times \mathcal{F}$: an option is $o = (P, f)$ for a peer $P \in \mathcal{P}$, and $f(o)$ denotes its facility. Throughout, "option" and "ingress" are synonyms for $o \in \mathcal{O}$. Sending one unit of traffic from U through o incurs penalty $\phi(o, U)$. Option o and facility f have capacities κ_o and κ_f. Opening facility f costs c_f, and serving traffic incurs per-unit operating costs b_o and b_f. At most B_{prefix} prefix identifiers can be used.

The joint model has three kinds of decisions:

Decision	Variables	Meaning
Deployment	$x_f, y_o \in \{0,1\}$	Open facility f; activate option o.
Announcements	$z_{o,k} \in \{0,1\}$	Option o announces prefix k.
Assignment	$\lambda_{U,k}, r_{U,o,k} \in [0,1]$	Fraction of U assigned prefix k; fraction of U served by o through k.

Write $a_o(U) = \sum_{k=1}^{B_{\text{prefix}}} r_{U,o,k}$ for the total fraction of group U served by option o. A fractional $\lambda_{U,k}$ splits one group across several prefixes. BGP gives each network a single best route per prefix, so such a split assumes the group can be steered at a finer grain, for example by handing different destination prefixes to parts of the group through DNS. If each group must use a single prefix, take $\lambda_{U,k} \in \{0,1\}$. We assume every group has a *strict, prefix-independent* preference ranking. The notation $o' \succ_U o$ means that U prefers o' to o; equivalently, $\pi_o(U)$ denotes a numerical preference score that increases with desirability. This assumption is essential to the graph-coloring analysis below; if rankings depend on the prefix, the graph must be defined differently. The model is the routing model of the main COMPASS paper in the special case where every group can reach every announcing ingress and rankings do not depend on the prefix; there, Lemma 1 below is that paper's exact-realization lemma.

2.2 Joint formulation

A natural objective is to minimize total routing penalty:

$$\min \sum_{U \in \mathcal{U}} \sum_{o \in \mathcal{O}} w_U \phi(o,U) a_o(U). \tag{1}$$

Its constraints are

$$\sum_{k=1}^{B_{\text{prefix}}} \lambda_{U,k} = 1 \qquad \forall U, \tag{2}$$

$$\sum_{o \in \mathcal{O}} r_{U,o,k} = \lambda_{U,k} \qquad \forall U, k, \tag{3}$$

$$r_{U,o,k} \le z_{o,k} \le y_o \le x_{f(o)} \qquad \forall U, o, k, \tag{4}$$

$$r_{U,o,k} \le 1 - z_{o',k} \qquad \forall U, k, o, o' \text{ with } o' \succ_U o, \tag{5}$$

$$a_o(U) = \sum_{k=1}^{B_{\text{prefix}}} r_{U,o,k} \qquad \forall U, o, \tag{6}$$

$$\sum_{U} w_U a_o(U) \le \kappa_o y_o \qquad \forall o, \tag{7}$$

$$\sum_{o: f(o)=f} \sum_{U} w_U a_o(U) \le \kappa_f x_f \qquad \forall f. \tag{8}$$

Equations (2)–(3) allocate all demand. Equation (4) permits traffic only on announcing, active, deployed options. Equation (5) is the crucial BGP-choice constraint: traffic from U cannot use o on prefix k if an option o' preferred by U also announces k. Equations (7)–(8) limit the loads induced by those winners.

The monetary deployment and operating cost is

$$C(x,y,a) = \sum_f c_f x_f + \sum_f b_f \sum_{o: f(o)=f} \sum_U w_U a_o(U) + \sum_o b_o \sum_U w_U a_o(U), \tag{9}$$

which may be constrained by a deployment budget $C(x,y,a) \le C_{\max}$. We assume every user can use every prefix identifier; otherwise, a prefix-availability constraint must also be imposed.

The problem contains two distinct combinatorial optimization problems. Even without BGP routing preferences, selecting capacitated facilities and assigning demand is NP-hard once deployment is limited by the budget $C_{\max}$ or charged opening costs c_f in the objective; with neither, opening every facility is optimal. Even for a fixed deployment and target assignment, determining the minimum number of prefixes required to realize that assignment exactly is equivalent to graph coloring, which is also NP-hard.

The second difficulty arises because prefix announcements couple routing decisions across users: adding an option to a prefix can redirect traffic from other options, potentially increasing routing cost or violating capacity constraints.

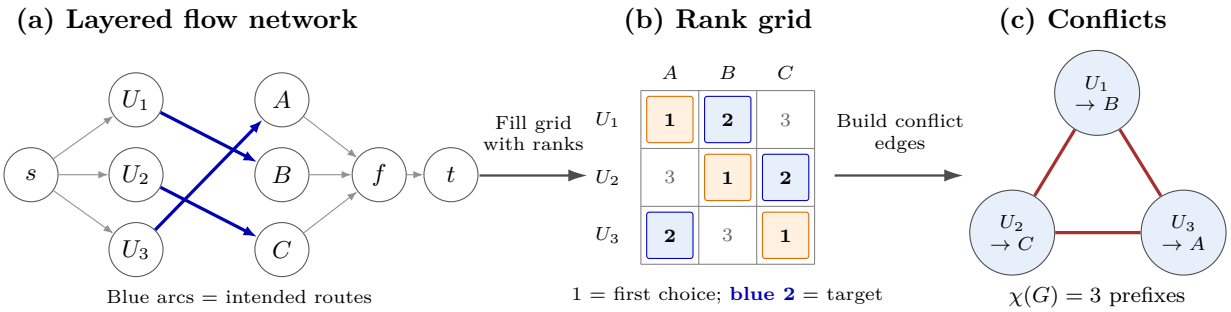


Figure 1: How a full-control solution becomes a conflict graph. (a) The chosen user–ingress flows are $U_1 \to B$, $U_2 \to C$, and $U_3 \to A$. (b) Each chosen arc identifies a *target* (blue cell); the user's independent preference ordering supplies the ranks (1 = most preferred). All three targets have rank 2. (c) Keep one vertex per target. An edge is added when one user's first choice is the ingress targeted by the other user. Every pair conflicts, so exact realization needs three prefixes.

2.3 A two-stage approach

Stage 1: select infrastructure under full control. Ignore the prefix limit and allow COMPASS to send arbitrary fractions $a_o(U)$ of each user group directly to active options. For any *fixed* deployment, this is a minimum-cost flow on

$$s \longrightarrow \mathcal{U} \longrightarrow \mathcal{O}_{\text{active}} \longrightarrow \mathcal{F}_{\text{open}} \longrightarrow t.$$

The source supplies w_U at each user, a $U \to o$ arc has unit cost $\phi(o, U)$, an $o \to f(o)$ arc has capacity κ_o, and an $f \to t$ arc has capacity κ_f. The per-unit operating costs b_o and b_f can be added to the corresponding arcs when they belong to the objective. When they appear only in the budget $C(x, y, a) \le C_{\max}$, moving that constraint into the objective with a Lagrange multiplier gives the same flow problem for each multiplier value; the opening costs c_f are fixed once the deployment is fixed. A feasible flow sending $\sum_U w_U$ units is the optimal divisible assignment for that *fixed* deployment. Selecting the deployment itself remains a discrete search.

Let $D \subseteq \mathcal{O}$ be the selected options and let $a^\star_o(U)$ be the resulting full-control target.

Stage 2: realize the target using prefixes. For each positive fraction $a^\star_o(U)$, introduce an *atomic intended assignment* $v = (U, o)$ with traffic weight $w_v = w_U a^\star_o(U)$. The word "atomic" refers only to the realization step: we treat this component as one item to be assigned to one prefix. If several components share a prefix, all their intended options announce that prefix. Each component is then routed to its most preferred announcing option. We ask how to partition these components into no more than B_{prefix} prefixes while preserving their intended routes, or minimizing the loss when exact preservation is impossible.

Figure 1 follows one toy assignment through these representations. In the full-control flow network, the three selected user-to-ingress arcs carry $U_1 \to B$, $U_2 \to C$, and $U_3 \to A$. The middle panel records these positive flows as filled cells in a user-by-ingress grid. It also marks each user's first choice, which is not the target in this example. The right-hand panel keeps only the selected assignments and joins two whenever sharing a prefix would cause one group to choose the other's ingress instead.

3 Exact realization is graph coloring

3.1 When do two intended assignments conflict?

Consider two assignment components $v = (U, o)$ and $v' = (U', o')$ with $o \neq o'$. If they use the same prefix, both o and o' must announce it. They can share that prefix *only if* U does not prefer o' over o and U' does not prefer o over o'. Thus we create a conflict edge whenever

$$v \sim v' \quad \Longleftrightarrow \quad o' \succ_U o \quad \text{or} \quad o \succ_{U'} o'. \tag{10}$$

Assignments targeting the same ingress do not conflict. Let G denote the resulting undirected *assignment conflict graph*: its vertices are the positive components of $a^\star$, not the ingress options themselves.

Why the example requires three prefixes. In Figure 1, each group is assigned its second choice: $A \succ_{U_1} B$, $B \succ_{U_2} C$, and $C \succ_{U_3} A$. Every pair of the three target assignments conflicts. For example, (U_1, B) and (U_3, A) cannot share a prefix: because A must announce that prefix for U_3, U_1 would choose A rather than B. Similarly, (U_2, C) conflicts with (U_1, B), and (U_3, A) conflicts with (U_2, C). This gives the triangle in panel (c).

Lemma 1 (Exact prefix realization)**.** *Assume strict, prefix-independent rankings and that a prefix announces exactly the distinct intended options of its assignment class. A partition of the assignment components preserves every intended route if and only if every class is an independent set of G. Consequently the minimum number of prefixes needed for exact realization is $\chi(G)$.*

Proof. If a class has no conflict edge, then no component prefers another component's intended option over its own. Every component therefore chooses its target among all options announced for that class. Conversely, if a class contains both endpoints of a conflict edge, at least one endpoint prefers the other's intended option and is redirected. A partition into independent sets is exactly a proper vertex coloring, with each color representing one prefix. □

Announcing a class's prefix at options beyond its intended ones never helps. Under strict rankings, an extra option either ranks below the target of every component in the class, and then no component uses it, or it ranks above some component's target and takes that component's traffic.

The conflict graph characterizes whether a prefix allocation preserves the target routing decisions rather than their total cost. Two different routes may incur the same penalty, so a prefix allocation can change the target assignments without changing the objective value. Throughout, we use "exact realization" to mean preserving the target assignments themselves, not only their total cost. This matters beyond ties: moving traffic to an option with the same penalty can still overload that option or its facility, violating κ_o or κ_f.

3.2 The simpler, but coarser, ingress graph

Let $A(v)$ denote the intended ingress of component v, and let

$$\mathcal{S} = \{A(v) : v \in V(G)\}, \qquad M = |\mathcal{S}|.$$

Coloring every component by its intended ingress is always valid; hence

$$\chi(G) \le M. \tag{11}$$

One may also build a *flattened ingress graph* F whose vertices are the ingresses in $\mathcal{S}$: join A and B if some component intended for A prefers B, or vice versa. Every coloring of F induces a valid coloring of G: give each component the color of its intended ingress. Components with the same intended ingress never conflict, and a conflict between a component intended for A and one intended for B is exactly an edge A–B of F, so no conflict edge of G lies inside a color class. This gives

$$\chi(G) \le \chi(F) \le M. \tag{12}$$

The reverse inequality need not hold, since coloring F forces *all* components assigned to the same ingress to share a color, whereas G may place them on different prefixes and use fewer colors overall. We keep these two graphs distinct in the proofs below.

4 Why assignment rank alone is not enough

For an intended assignment (U, A), let $r_U(A)$ be the rank of A in U's strict ordering (rank 1 is best; the score $\pi_o(U)$ of Section 2 runs the other way, with larger values more preferred, as in the main COMPASS paper), Only ingresses that are actually used can redirect a group, so we count rank among them. Assuming at least one positive assignment component (so $M \ge 1$), define the *active rank*

$$k_U(A) = 1 + \big|\{B \in \mathcal{S} : B \succ_U A\}\big| \le r_U(A), \qquad K = \max_{(U,A)\in V(G)} k_U(A),$$

where $\mathcal{S}$ is the set of ingresses used by the target (Section 3). Thus $K - 1$ is the largest number of used ingresses that any group ranks above its target. The full rank $r_U(A)$ also counts ingresses that no group is assigned to, and on measured deployments it can be far larger; Section 5 uses it only for global agreement, where we write $\bar{K} = \max_{(U,A)\in V(G)} r_U(A) \ge K$.

If $K = 2$, each user prefers at most one used ingress to its target. It is tempting to conclude that few prefixes should suffice. The triangle already demonstrates that three prefixes can be necessary for $K = 2$. The next construction shows that there is *no bound at all* that depends only on K.

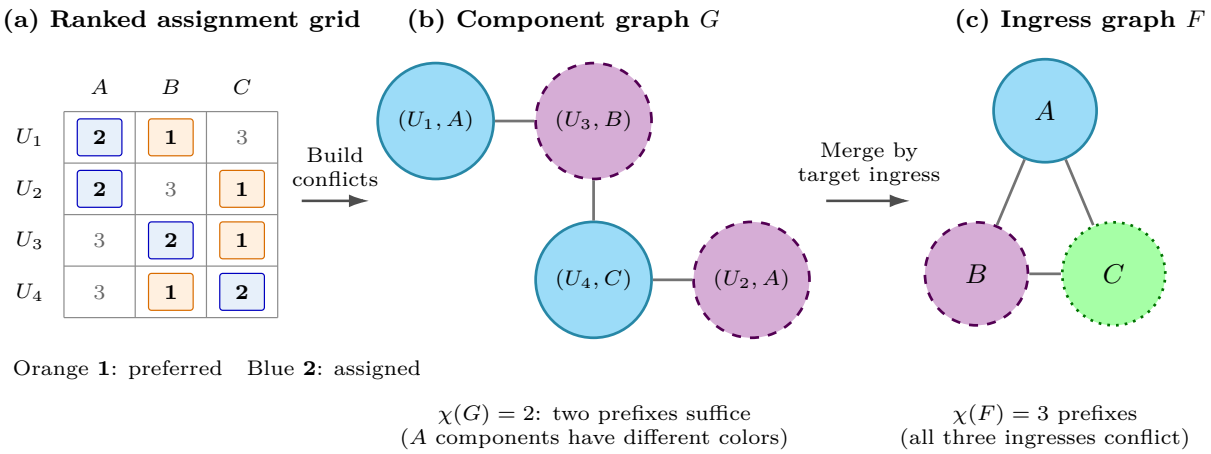


Figure 2: Flattening the conflict graph can increase the required number of prefixes. (a) Every user is assigned to its second-choice ingress (blue **2**); its first choice is shown in orange and its third choice in gray. (b) The four assignment components form the path (U_1, A)–(U_3, B)–(U_4, C)–(U_2, A), which can be colored using two prefixes. Vertex fill and outline style (solid, dashed, dotted) indicate prefix assignments. In particular, the two components assigned to A use different prefixes. (c) Merging all components with the same intended ingress produces the flattened ingress graph F: the three pairwise conflicts A–B, B–C, and A–C form a triangle, requiring three colors. Thus $\chi(G) = 2 < \chi(F) = 3$.

4.1 The shift-graph construction

Take n ingresses $A_1, \ldots, A_n$. For every pair $1 \le i < j \le n$, create one user U_{ij} assigned to A_j, with A_i ranked first and A_j second; all other ingresses rank below A_j. Represent its assignment vertex by (i, j).

A vertex (i, j) conflicts with (h, i) whenever $h < i$: the user U_{ij} prefers A_i, which is the intended ingress of (h, i). Thus the conflict graph has exactly the edges

$$V(G_n) = \{(i, j) : i < j\}, \qquad E(G_n) = \big\{\{(h, i), (i, j)\} : h < i < j\big\}. \tag{13}$$

Figure 3 gives the small $n = 5$ example. Unlike the three-user triangle, this graph is triangle-free; nevertheless, the highlighted odd cycle shows it cannot be colored with only two colors.

Theorem 2 (Second choices can require arbitrarily many prefixes)**.** *For every $n \ge 2$, there is an instance in which every user is assigned to its second-choice ingress, yet*

$$\chi(G_n) = \lceil \log_2 n \rceil.$$

Therefore no upper bound on the number of prefixes required for exact realization can depend only on K.

Proof. Use the construction above, where every intended ingress has rank 2. Suppose G_n has a proper coloring with c colors. For each ingress index j, collect the colors of the vertices assigned to A_j:

$$C_j = \{\mathrm{col}(i, j) : i < j\} \subseteq \{1, \ldots, c\}, \qquad C_1 = \varnothing.$$

The key observation is that these color sets must be *different for every two ingress indices*. To see why, fix $i < j$. Vertex (i, j) is adjacent to every vertex (h, i), $h < i$, by Equation (13). Its color is therefore absent from C_i. But (i, j) is itself assigned to A_j, so its color belongs to C_j. Hence $C_i \ne C_j$.

In routing terms, C_j is exactly the set of prefixes announced at A_j: under the announcement model of Lemma 1, a prefix is announced precisely at the intended ingresses of the components placed on it. The observation then says that to keep U_{ij} on A_j, its prefix must be announced at A_j but not at A_i, which U_{ij} prefers. No two ingresses may therefore announce the same set of prefixes. Note the order of quantifiers: the instance is fixed, and the argument applies to *every* proper coloring of G_n.

We have produced n distinct subsets $C_1, \ldots, C_n$ of a palette of c colors. Such a palette admits only 2^c distinct subsets. Consequently $n \le 2^c$, or $c \ge \lceil \log_2 n \rceil$.

For the matching upper bound, color vertex (i, j) by the lowest bit position at which $i - 1$ has a 0 and $j - 1$ has a 1; such a position exists because $i < j$. If (h, i) and (i, j) are adjacent, then $i - 1$ has a 1 at the color of (h, i) and a 0 at the color of (i, j), so the two colors differ. Every color is a bit position of a number below n, so $\lceil \log_2 n \rceil$ colors suffice. (G_n is the shift graph of Erdős and Hajnal.) Since n can grow without changing $K = 2$, no function of K alone bounds $\chi(G_n)$. □

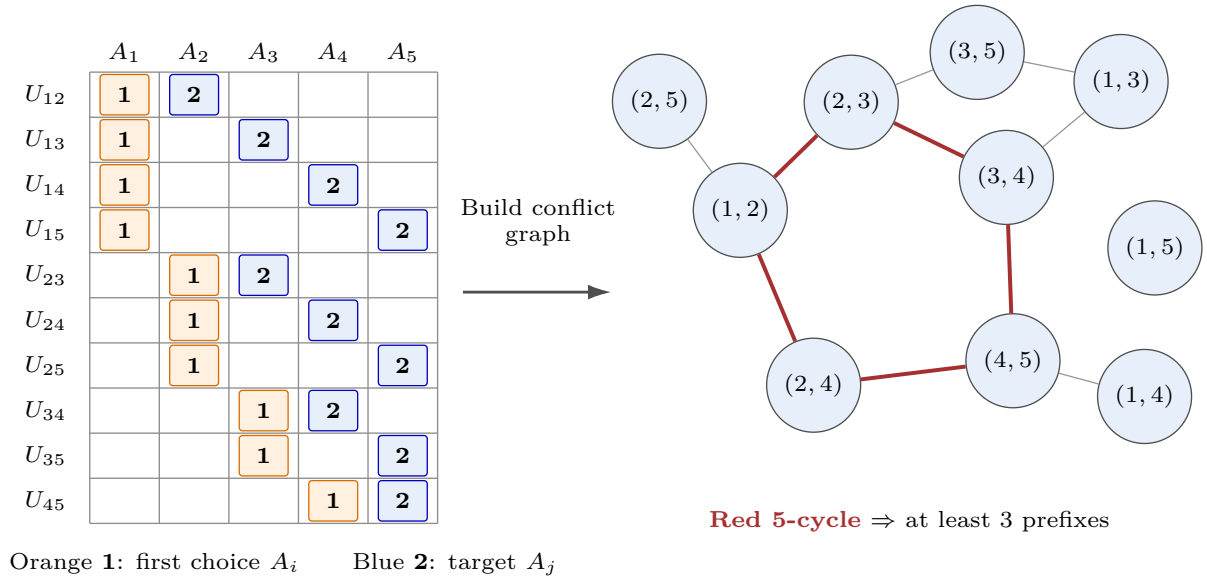


Figure 3: The shift-graph construction, using the same visual rules as Figure 1. For each $i < j$, user U_{ij} ranks A_i first and targets A_j second; blank cells represent lower-ranked ingresses. Each blue target cell becomes vertex (i, j). Since U_{ij} prefers A_i to its assigned ingress, vertex (i, j) conflicts with every vertex (h, i) assigned to A_i. The red five-cycle shows that G_5 cannot be 2-colored even though every target has rank 2; gray edges are the remaining conflicts. Three prefixes suffice, so $\chi(G_5) = 3 = \lceil \log_2 5 \rceil$.

The problem is *collective compatibility*, not the number of better choices available to any single user. Even with one better alternative per user, those alternatives can interact across many assignments and force increasingly many colors.

5 What preference agreement can guarantee

The impossibility theorem says that K alone is insufficient. It does not say that every realistic preference profile needs many prefixes. We now identify two distinct properties that give useful upper bounds: *global agreement* about which ingresses appear among users' top choices, and *local agreement* about which alternatives are preferred by users assigned to the same ingress.

5.1 Global agreement: how many different ingresses can be targets?

Recall that $\mathcal{S}$ is the set of ingresses used by the target assignment and $M = |\mathcal{S}|$. By Equation (11), M prefixes always suffice: assign one prefix to all components targeting each distinct ingress.

Proposition 3 (Identical rankings)**.** *If all users share the same strict ranking and every target has active rank at most K, then*

$$\chi(G) = M \leq K.$$

Proof. In the common ranking, the lowest-ranked intended ingress has the other $M - 1$ intended ingresses above it, so its active rank is M and $M \leq K$. Now take one assignment vertex for each distinct intended ingress. For any two different ingresses, a user assigned to the lower-ranked one prefers the higher-ranked one. The chosen vertices therefore form an M-clique, giving $\chi(G) \geq M$. Together with $\chi(G) \leq M$, this proves equality. □

Identical rankings are stronger than necessary to obtain an upper bound in terms of rank. The remaining global results count rank among *all* ingresses, so they use the full rank $\bar{K}$ of Section 4. If every user's top-$\bar{K}$ *set* is the same set T of size $\bar{K}$, users may disagree on its internal order, but every target still belongs

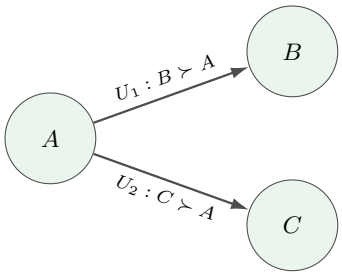


Figure 4: Local preference diversity aggregates *across* users with the same target. Each user assigned to A has only one better alternative ($K = 2$), but the union of their better assigned alternatives has size two. Here B and C are assumed to be targets of other users, so $B, C \in \mathcal{S}$. The theorem bounds the outdegree of this directed ingress graph, not the outdegree of a user-assignment vertex.

to T. Thus $M \le \bar{K}$ and $\chi(G) \le \bar{K}$. Equality need not hold because disagreements about the order can eliminate conflicts.

More generally, partition the user groups into *preference types*: groups of the same type share the same ranking, or at least the same set of top $\bar{K}$ ingresses. Let τ be the number of types and T_t the top-$\bar{K}$ set of type t, and define the *top-$\bar{K}$ diversity*

$$M_{\bar{K}} = \left| \bigcup_{t=1}^{\tau} T_t \right|.$$

Every intended ingress occurs in some user's top-$\bar{K}$ set; hence

$$\boxed{\chi(G) \le M \le M_{\bar{K}} \le \tau \bar{K}.} \tag{14}$$

This single observation includes the common-set case ($M_{\bar{K}} = \bar{K}$) and the case of two preference types. For two types with $|T_1 \cap T_2| = m$,

$$\chi(G) \le |T_1 \cup T_2| = 2\bar{K} - m. \tag{15}$$

The more their sets overlap, the fewer distinct target ingresses can appear. These are upper bounds, not exact prefix counts: the target ingresses may be numerous without all conflicting.

5.2 Local agreement: how many better alternatives compete with an ingress?

The bound $\chi(G) \le M$ can be loose if COMPASS uses many ingresses whose assignments can nevertheless share prefixes. We need a measure of how their preferences interact.

For each assigned ingress $A \in \mathcal{S}$, define its set of *aggregated better assigned alternatives*

$$H(A) = \bigcup_{v=(U,A)} \{B \in \mathcal{S} : B \succ_U A\}, \qquad R = \max_{A \in \mathcal{S}} |H(A)|. \tag{16}$$

Only alternatives that are themselves assigned somewhere in the target matter: if no component needs ingress B to be announced, the assignment can omit B from every prefix.

The distinction between K and R is crucial. When every target has active rank at most K, an individual user prefers at most $K - 1$ used alternatives. But *different users assigned to the same ingress* may prefer different alternatives. Their union $H(A)$ may therefore be much larger than $K - 1$.

Figure 4 shows the aggregation required by the proof. It is deliberately drawn on *ingresses* rather than on individual assignment components: the next theorem colors the flattened graph F.

Theorem 4 (Bounded local preference diversity)**.** *Recall that $M = |\mathcal{S}|$ is the number of ingresses used by the target assignment. If $|H(A)| \le R$ for every assigned ingress A, then*

$$\boxed{\chi(G) \le \min\{M, 2R + 1\}.}$$

Proof. The hypothesis bounds how many ingresses the users of each ingress would defect to, not how many ingresses conflict with it: in F an ingress can have arbitrarily high degree. We therefore keep track of direction. Construct the directed graph $\vec{F}$ on the assigned ingresses $\mathcal{S}$, with an arc $A \to B$ whenever $B \in H(A)$, i.e., whenever some component intended for A prefers B. Every vertex has outdegree at most R. Ignoring the directions gives exactly the flattened conflict graph F of Section 3.

Consider any nonempty subset $X \subseteq \mathcal{S}$. Its induced directed subgraph contains at most $R|X|$ arcs: each vertex in X has at most R outgoing arcs, including those leaving X. Every undirected edge of $F[X]$ is represented by at least one arc, so

$$|E(F[X])| \leq R|X|.$$

The average degree in $F[X]$ is thus at most $2R$. Every nonempty induced subgraph of F therefore contains a vertex of degree at most $2R$. Removing such a vertex repeatedly, and coloring in reverse removal order, uses at most $2R+1$ colors. Hence $\chi(F) \leq 2R+1$. Finally, Equation (12) gives $\chi(G) \leq \chi(F) \leq 2R+1$, and Equation (11) also gives $\chi(G) \leq M$. □

This is the correct form of the bounded-outdegree argument: the relevant outdegree is aggregated at the *ingress* level. A user-level outdegree of at most $K-1$ does not imply the ingress-level bound $R \leq K-1$; the shift-graph construction demonstrates why. There, every target has rank 2, yet $H(A_j) = \{A_1, \ldots, A_{j-1}\}$, so $R = n-1$ and Theorem 4 gives only $\chi(G_n) \leq n$. This is consistent with Theorem 2: the bound is valid but uninformative when users assigned to the same ingress prefer many different alternatives. The bound $2R+1$ is tight for $\chi(F)$: with $R=1$ the ingress graph can be an odd cycle, and in general a regular tournament on $2R+1$ ingresses has every outdegree equal to R and a complete underlying graph, so $\chi(F) = 2R+1$. The true $\chi(G)$ can still be smaller.

The theorem bounds $\chi(F)$, i.e., the number of prefixes needed when all components assigned to an ingress share one prefix. Since $\chi(F)$ can exceed $\chi(G)$ (Figure 2), $\min\{M, 2R+1\}$ is a sufficient prefix budget, not the exact number of prefixes required.

5.3 When does the $2K-1$ bound hold?

Corollary 5 (Local agreement)**.** *Suppose every target has active rank at most K and all users assigned to the same ingress A agree on the set of ingresses ranked above A. Then*

$$\chi(G) \leq \min\{M, 2K-1\}.$$

Proof. For each A, the common better-choice set contains at most $K-1$ used ingresses, so $|H(A)| \leq K-1$. Apply Theorem 4. □

This does *not* require a global ranking shared by all users. Users assigned to different ingresses may disagree completely; what matters is that users with a common intended ingress do not introduce new competing alternatives into the union $H(A)$. This is precisely the hypothesis that the shift construction violates: there, each user assigned to A_j prefers a different ingress.

We can weaken exact agreement. Choose one reference user U_A for each assigned ingress A and set

$$H_0(A) = \{B \in \mathcal{S} : B \succ_{U_A} A\}.$$

If every target has active rank at most K and, for every A,

$$|H(A) \setminus H_0(A)| \leq \Delta,$$

then $|H(A)| \leq K-1+\Delta$. Hence

$$\boxed{\chi(G) \leq \min\{M, 2K+2\Delta-1\}.} \tag{17}$$

The parameter Δ counts *distinct additional ingress labels*, not the number of users who disagree. It can remain small even when many users have different complete rankings.

Alternatively, suppose at most τ_{loc} distinct preference types occur among users assigned to any single ingress. Each type contributes at most $K-1$ better alternatives, so $R \leq \tau_{\text{loc}}(K-1)$ and

$$\boxed{\chi(G) \leq \min\{M, 2\tau_{\text{loc}}(K-1)+1\}.} \tag{18}$$

Unlike the global bound $\tau\bar{K}$, this guarantee depends on how preferences vary *within each assigned ingress*, not on the number of preference types in the population as a whole.

5.4 Setting outlier ingresses aside

The parameter R is a maximum over all assigned ingresses. A single ingress whose users have many different better alternatives, such as a catch-all site that receives traffic from many regions, can therefore make Theorem 4 uninformative even when every other ingress has few competitors. Giving such ingresses prefixes of their own removes them from the maximum.

Corollary 6 (Setting outlier ingresses aside)**.** *For any $S \subsetneq \mathcal{S}$, let*

$$R_S = \max_{A \in \mathcal{S} \setminus S} |H(A) \setminus S|.$$

Then

$$\chi(G) \le \chi(F) \le |S| + 2R_S + 1, \qquad \textit{hence} \qquad \boxed{\chi(G) \le \min\Big\{M,\ \min_{S \subsetneq \mathcal{S}}\big(|S| + 2R_S + 1\big)\Big\}.}$$

Proof. Give each ingress in S a color of its own, used by no other ingress; then no edge of F incident to S joins two vertices of the same color. The induced subgraph $F - S$ is the flattened graph of the remaining ingresses, and in the directed graph $\vec{F} - S$ each vertex A has outdegree $|H(A) \setminus S| \le R_S$. The argument of Theorem 4 applied to $F - S$ colors it with at most $2R_S + 1$ further colors, so $\chi(F) \le |S| + 2R_S + 1$. Equation (12) gives $\chi(G) \le \chi(F)$. Taking $S = \varnothing$ recovers Theorem 4. □

For example, if one ingress X has $|H(X)| = 50$ while every other ingress has $|H(A)| \le 2$, Theorem 4 gives 101 prefixes, whereas setting X aside gives $1 + 2 \cdot 2 + 1 = 6$. Setting an ingress aside pays off when it is an outlier in *outdegree*, i.e., when its own users disagree. A popular ingress that appears in many sets $H(A)$ lowers R by at most one when removed, so setting it aside saves at most one prefix.

5.5 Summary of the structural results

The results describe different forms of preference agreement:

Assumption	**Exact-prefix guarantee**
Identical strict rankings	$\chi(G) = M \le K$
Common top-$\bar{K}$ set (full rank)	$\chi(G) \le M \le \bar{K}$
Top-$\bar{K}$ diversity $M_{\bar{K}}$	$\chi(G) \le M \le M_{\bar{K}}$
Local better-choice diversity R	$\chi(G) \le \min\{M, 2R + 1\}$
Local agreement among users assigned to each ingress	$\chi(G) \le \min\{M, 2K - 1\}$
Few outlier ingresses S set aside	$\chi(G) \le \min\{M, \lvert S\rvert + 2R_S + 1\}$
Common weak order with ties of size at most w	$\chi(G) \le \min\{M, 2K + 2w - 3\}$
Arbitrary rankings	No bound depending only on K
Arbitrary rankings, targets of active rank at most K	$\chi(G) \le L_{K-1}(M) = O(K^2 \log M)$
Any instance	$\lceil \log_2 \chi(F) \rceil \le \chi(G) \le \chi(F)$

In particular, global agreement keeps the *number of distinct target ingresses* small. Local agreement permits many target ingresses but keeps the *conflict structure* sparse enough to color with fewer prefixes. Neither parameter necessarily equals the true chromatic number.

A common partial order does not, by itself, rescue a rank-only guarantee. If it leaves all ingresses used in the shift construction incomparable, the users' strict rankings can extend that partial order while still generating G_n with unbounded chromatic number. A useful hypothesis must restrict the comparisons that actually induce conflict edges. A common weak order does so when its ties are small: suppose every user's ranking extends a ranking of the ingresses into tied buckets, each of at most w ingresses. An ingress B in a strictly higher bucket than A is ranked above A by every user, so a user assigned to A at active rank at most K sees at most $K - 1$ of them; the remaining members of $H(A)$ share A's bucket. Hence $|H(A)| \le K + w - 2$, and Theorem 4 gives $\chi(G) \le \min\{M, 2K + 2w - 3\}$. With $w = 1$ the order is total and this recovers the $2K - 1$ bound.

6 How tight are these bounds?

The previous sections give upper bounds that rest on agreement, and Theorem 2 shows that rank alone cannot bound the number of prefixes independently of the instance. Two further results complete the picture. The ingress-level plan of Equation (12) is never more than exponentially worse than the best plan, and rank together with the number of used ingresses always gives a logarithmic bound.

6.1 Flattening costs at most an exponential

Proposition 7 (Flattening sandwich)**.** *For every instance,*

$$\lceil \log_2 \chi(F) \rceil \;\le\; \chi(G) \;\le\; \chi(F).$$

Proof. The upper bound is Equation (12). For the lower bound, fix a proper coloring of G with $c = \chi(G)$ colors, and give each used ingress A the set C_A of colors used by components intended for A. Let A and B be adjacent in F; say some component (U, B) has $A \succ_U B$. Its color belongs to C_B. It cannot belong to C_A, because a component intended for A with that color would conflict with (U, B). Hence $C_A \neq C_B$, so $A \mapsto C_A$ is a proper coloring of F with at most 2^c distinct sets, and $\chi(F) \le 2^{\chi(G)}$. □

In routing terms, C_A is the set of prefixes announced at A: any two ingresses joined in F must announce different sets of prefixes. The proof of Theorem 2 is this argument applied to the shift instance, where F is complete on the $n-1$ used ingresses. Both inequalities can be tight: identical rankings give $\chi(G) = \chi(F) = M$ (Proposition 3), and the shift instance has $\chi(F) = n-1$ but $\chi(G_n) = \lceil \log_2 n \rceil$, so the lower bound is attained up to rounding.

6.2 What rank does guarantee

For $d \ge 0$, call a list of sets *d-cover-free* if no member is contained in the union of d other members, and let $L_d(M)$ be the smallest L for which $\{1, \dots, L\}$ has a d-cover-free list of M subsets. Since the M singletons form such a list, $L_d(M) \le M$.

Theorem 8 (Rank gives a logarithmic bound)**.** *If every target has active rank at most K, then*

$$\chi(G) \le L_{K-1}(M).$$

For $K = 2$ this is $\chi(G) \le \min\{L : \binom{L}{\lfloor L/2 \rfloor} \ge M\} = \log_2 M + O(\log\log M)$, and in general $\chi(G) = O(K^2 \log M)$.

Proof. Let $L = L_{K-1}(M)$ and give the used ingresses codes $c(A) \subseteq \{1, \dots, L\}$ forming a $(K-1)$-cover-free list. A component (U, A) ranks at most $K-1$ used ingresses above A; call them $\mathcal{B}$. Because no code is covered by $K-1$ others, $c(A) \setminus \bigcup_{B \in \mathcal{B}} c(B)$ is nonempty: place (U, A) on any prefix p in it. Every component on prefix p is intended for an ingress whose code contains p. If two components (U, A) and (U', B) on prefix p conflicted, say with $B \succ_U A$, then $p \in c(B)$, contradicting the choice of p for (U, A). So the placement is a proper coloring of G with L colors.

For $K = 2$, a 1-cover-free list is an antichain of distinct sets, and by Sperner's theorem the largest antichain of subsets of $\{1, \dots, L\}$ has $\binom{L}{\lfloor L/2 \rfloor}$ members. For general d, known constructions of d-cover-free families (Erdős, Frankl and Füredi) give $L_d(M) = O(d^2 \log M)$. □

In routing terms, $c(A)$ is the set of prefixes ingress A may announce. A group is placed on a prefix that its target announces and that none of the ingresses it prefers announces, so the code length, not the number of ingresses, is what limits the prefix count. Unlike Theorem 4, this bound colors G directly: components intended for the same ingress may use different prefixes. Together with Theorem 2, it shows that for second-choice assignments the number of prefixes grows logarithmically in the number of ingresses: some instances need $\lceil \log_2 n \rceil$, and none needs more than about $\log_2 M$.

7 What if the prefix budget is too small?

When $B_{\text{prefix}} < \chi(G)$, no partition can preserve every target route. The correct objective is then the cost of the *actual winners*, not simply how many conflict edges lie inside each prefix class.

7.1 Why pairwise edge penalties can mislead

Suppose component 1 targets o_1 but ranks $o_3 \succ o_2 \succ o_1$. Relative to the intended route, moving this component to o_2 costs 4 and moving it to o_3 costs 6. Another component, component 3, targets o_3 but prefers o_4, at additional cost 5. Assume the other relevant preference comparisons are nonconflicting and all intended costs are zero. The conflict edges have proxy weights $(1,2):4$, $(1,3):6$, and $(3,4):5$.

With two prefixes, consider

Partition	Prefix classes	Sum of internal edge penalties	Actual cost
(i)	$\{1,2,3\},\ \{4\}$	$4+6=10$	6
(ii)	$\{1,2\},\ \{3,4\}$	$4+5=9$	$4+5=9$

The edge proxy chooses partition (ii), but the actual cost favors (i). In (i), component 1 sees both o_2 and o_3 and chooses *only* o_3. Its two incident edges describe alternative redirections, not two costs that can be added. The class's entire announcement set determines its winner.

7.2 Exact fixed-target formulation

Let V be the atomic assignment components. For any nonempty class $S \subseteq V$, define

$$\mathcal{A}(S) = \{o_v : v \in S\}, \qquad h_v(S) = \arg\max_{o \in \mathcal{A}(S)} \pi_o(U_v),$$

where arg max selects the most-preferred option in the announced set. Define its realized routing cost, regret relative to the target, and induced option/facility loads:

$$g(S) = \sum_{v \in S} w_v \phi(h_v(S), U_v), \tag{19}$$

$$\rho(S) = \sum_{v \in S} w_v \big[\phi(h_v(S), U_v) - \phi(o_v, U_v)\big], \tag{20}$$

$$\ell_o(S) = \sum_{v \in S:\, h_v(S) = o} w_v, \tag{21}$$

$$\ell_f(S) = \sum_{o : f(o) = f} \ell_o(S). \tag{22}$$

(The regret $\rho(S)$ can be negative if an induced winner is cheaper than the full-control target for that component; the fixed target's total cost is constant in either case.)

Let $\mathcal{C} \subseteq 2^V \setminus \{\varnothing\}$ be candidate prefix classes, and let $\theta_S \in \{0,1\}$ indicate whether class S receives a prefix. The fixed-target realization problem is

$$\begin{aligned}
\min_{\theta} \quad & \sum_{S \in \mathcal{C}} \rho(S)\theta_S \\
\text{s.t.} \quad & \sum_{S \ni v} \theta_S = 1 && \forall v \in V, \\
& \sum_{S \in \mathcal{C}} \theta_S \le B_{\text{prefix}}, \\
& \sum_{S \in \mathcal{C}} \ell_o(S)\theta_S \le \kappa_o && \forall o \in D, \\
& \sum_{S \in \mathcal{C}} \ell_f(S)\theta_S \le \kappa_f && \forall f, \\
& \theta_S \in \{0,1\} && \forall S \in \mathcal{C}.
\end{aligned} \tag{23}$$

The first two constraints partition all components into at most B_{prefix} prefix classes. The remaining constraints check capacities *after* preference-induced redirection. Choosing all nonempty subsets makes this an exact, but exponential-size, set-partitioning formulation for the fixed target and the class-announcement policy. If arbitrary prefix classes are generated selectively, optimality holds only over the generated family.

7.3 Practical approximation

A scalable implementation can start by assigning vertices to at most B_{prefix} colors using conflict weights as an inexpensive heuristic. But after assigning the colors, it must reconstruct each color's set of announced options and recompute: (i) every component's actual winner, (ii) option and facility loads, and (iii) the true realized objective. Candidate moves include recoloring one component, swapping two colors, or moving all components targeting one ingress. A move should be accepted only if it preserves capacity feasibility and improves the *actual* objective. If the goal is exact coloring and the prefix budget is not binding, DSATUR is a useful baseline.

The two-stage decomposition is not globally optimal for a fixed prefix budget. A deployment with a slightly worse full-control assignment may be much easier to realize and have a better final outcome. One extension is to incorporate a prefix-realization penalty into the infrastructure search; another is to jointly optimize deployment and prefix classes.

8 Implications for COMPASS

The mathematical results suggest concrete diagnostics for an empirical evaluation. For every full-control target, COMPASS can measure the maximum active rank K, the number M of distinct targeted ingresses, the top-$\bar{K}$ diversity $M_{\bar{K}}$, the aggregated local diversity R, and the observed number of colors needed for exact realization. These measurements distinguish two explanations for small prefix budgets: either the assignment uses few ingresses, or it uses many ingresses whose conflicts permit substantial color reuse.

Preference rankings may change faster than physical deployment. COMPASS can first attempt to recolor a fixed deployment when rankings change, and trigger deployment optimization only when the resulting routes are too costly or violate capacity.

The main practical quantity is the *realization gap*: how much the best available prefix scheme differs from the full-control target under the same infrastructure and capacities. Prefix realization is best understood as constrained menu design: the planner controls which options are visible together; each user chooses its preferred visible option. Rank by itself does not predict how many such menus are necessary. What matters is how the intended ingresses and the preferences for competing ingresses interact across the entire assignment.