CSE/ECE 848 Introduction to Evolutionary Computation

Module 4, Lecture 18, Part 2
More Combinatorial Optimization

Erik D. Goodman, Executive Director
BEACON Center for the Study of Evolution in Action
Professor, ECE, ME, and CSE

Newer Approaches to Evolutionary Combinatorial Optimization

- So far, all the methods we discussed to address combinatorial problems were stochastic: crossover and mutation occurred at random loci, and most offspring are worse than their parents
- Newer work by Whitley, et al., uses crossover that jumps from a pair of local optima to another, deterministically, "tunneling" among fitness peaks, generating the BEST of 2^K possible offspring, in O(N) time, that are at least piecewise locally optimal

Newer Approaches to Evolutionary Combinatorial Optimization

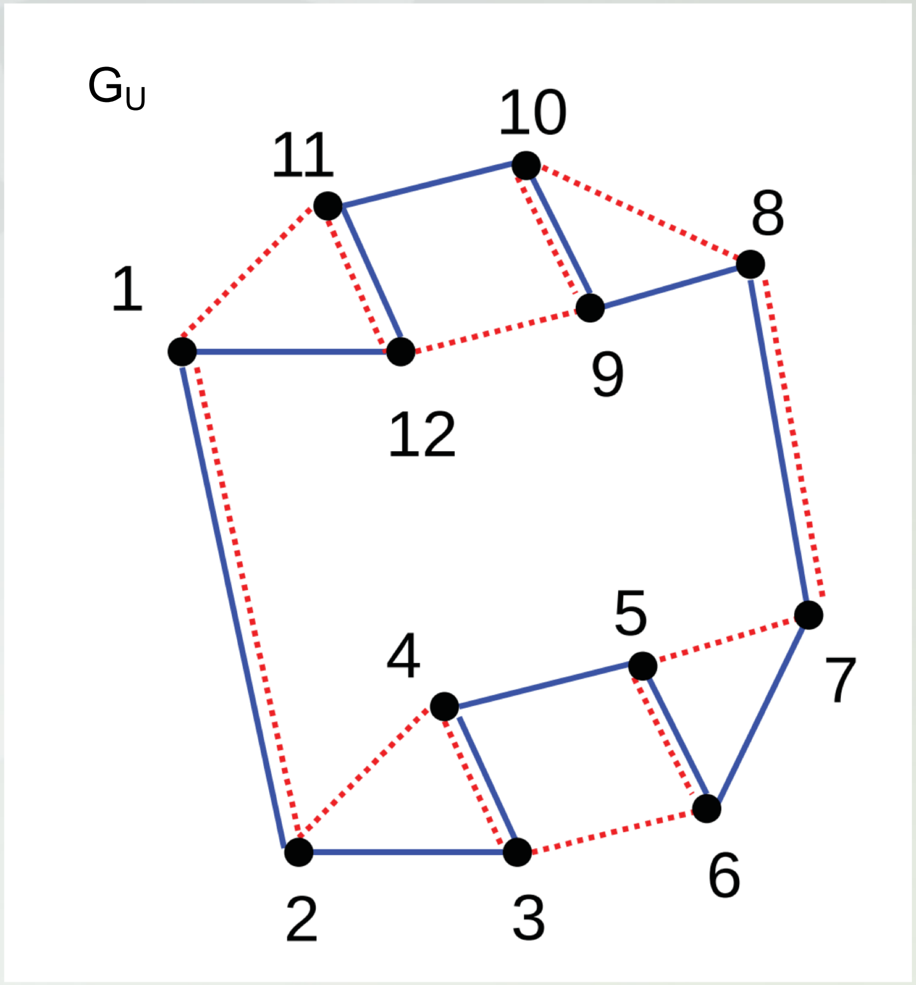
- We will just scratch the surface of these methods, but if you want to work seriously on combinatorial problems like TSP or kSAT, you should study the papers dealing with these topics
- A recent one is " New Generalized Partition Crossover for the Traveling Salesman Problem: Tunneling between Local Optima." Renato Tinos, Darrell Whitley and Gabriela Ochoa, *Evolutionary Computation,* **28**(2), 2020. It cites many more good papers.
- Most of the material and all of the example illustrations in this lecture are from that paper.
- Generalized Partition Crossover (GPX) enables efficient, deterministic linking of locally optimal subpaths (subgraphs) to produce longer, locally optimal subpaths

Properties of GPX

- All partition crossover operators are “respectful” and “transmit alleles”
- In “respectful” recombination, all common features (edges, in the TSP case) found in **both** parents are always inherited by the offspring
- Offspring generated by operators that “transmit alleles” are composed only of features (edges) contained in the parents
- (An operator that “transmits alleles” cannot introduce a new edge not found in either parent)
- These properties are not possessed by many other recombination operators for the TSP

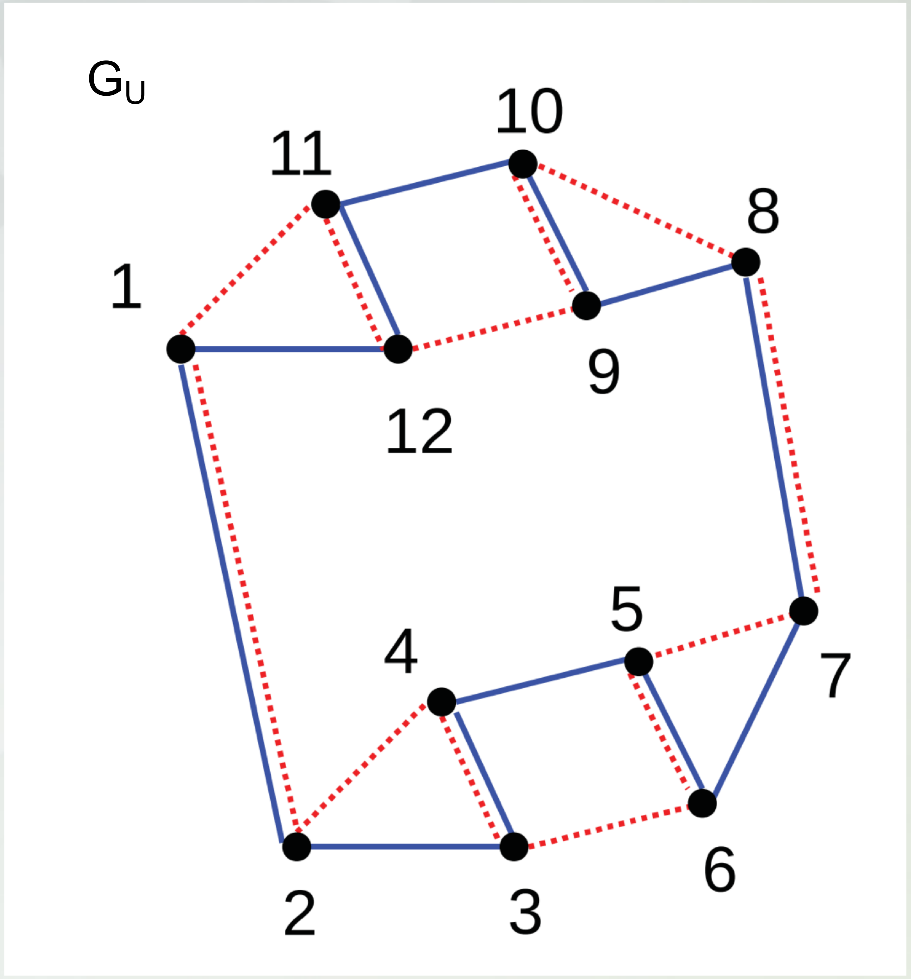
Illustrating Generalized Partition Crossover

- First step is to form the union graph G_U of the two subgraphs: dotted red edges are one parent; solid blue are the other
- Notice that some edges appear in both parents—watch those… remember, GPX is "respectful"…



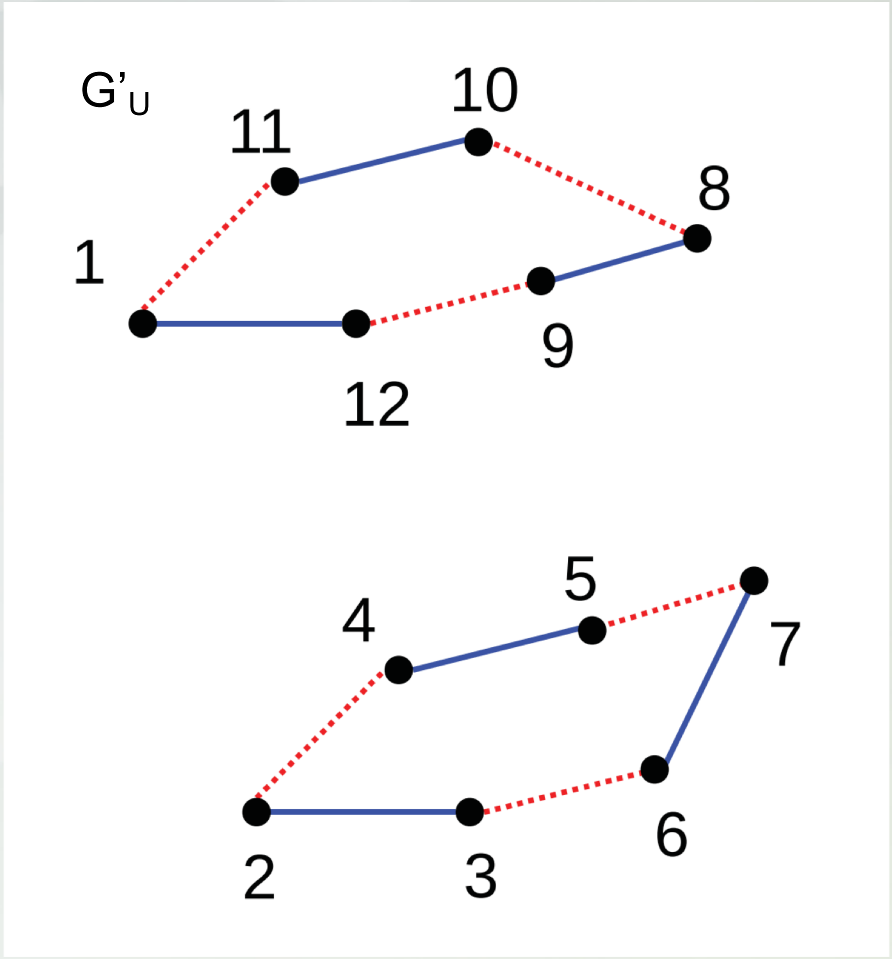
Illustrating Generalized Partition Crossover (GPX)

- The offspring inherits those *common* edges (red and blue), and they can be removed to leave the next graph… call this G'_U …
- (We also remove ALL vertices of degree 2 in the union graph… their incident edges match in both parents, so are inherited by the offspring, but there are none here.)



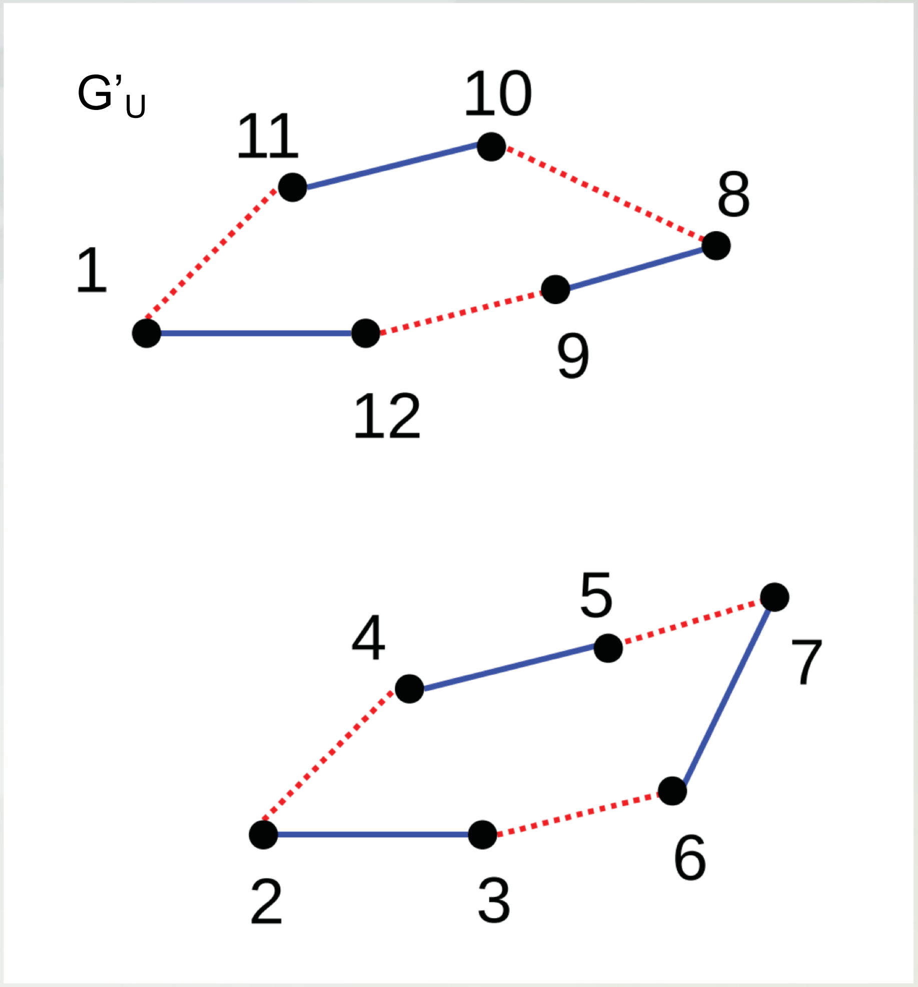
GPX (cont.)

- That leaves these two connected subgraphs
- Now we need some definitions before the next step
- A ***candidate component*** is made up of one or more connected subgraphs of G'_U
- We see two candidate components here



GPX (cont.)

- A ***portal*** is a vertex in a candidate component that connects (in G_U) to another vertex in a different candidate component
- Thus, portals exist as pairs, v_i and v_j. Here, (1, 2) and (7, 8) are pairs of portals—they connected these candidate components in the original G_U



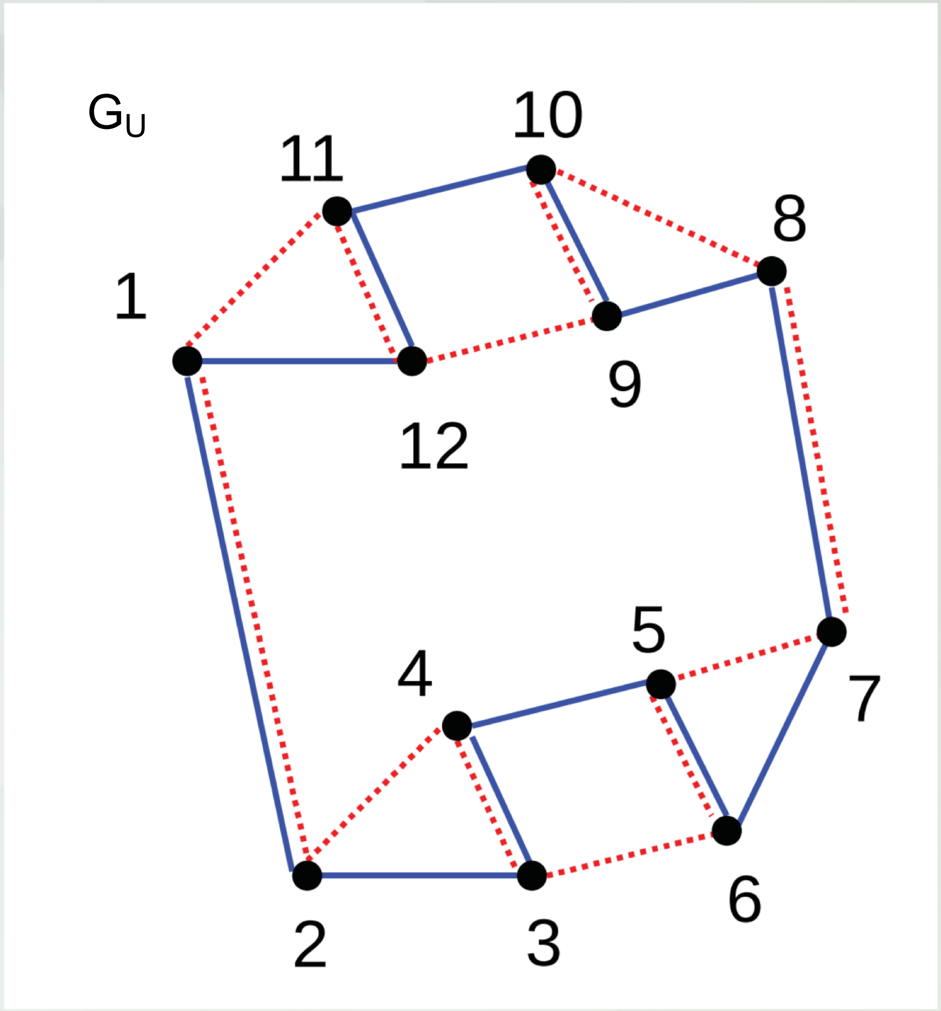
GPX (cont.)

- A ***recombining component*** of graph G_U is a candidate component of G'_U such that:

1. It contains z vertices, where 2x vertices are portals that connect to other recombining components by common edges, and the remaining z − 2x vertices only connect to vertices inside the recombining component;
2. There exists a traversal of the two parent permutations over the vertices in graph G_U such that both parents enter the recombining component at exactly the same portals and then exit the recombining component at exactly the same portals

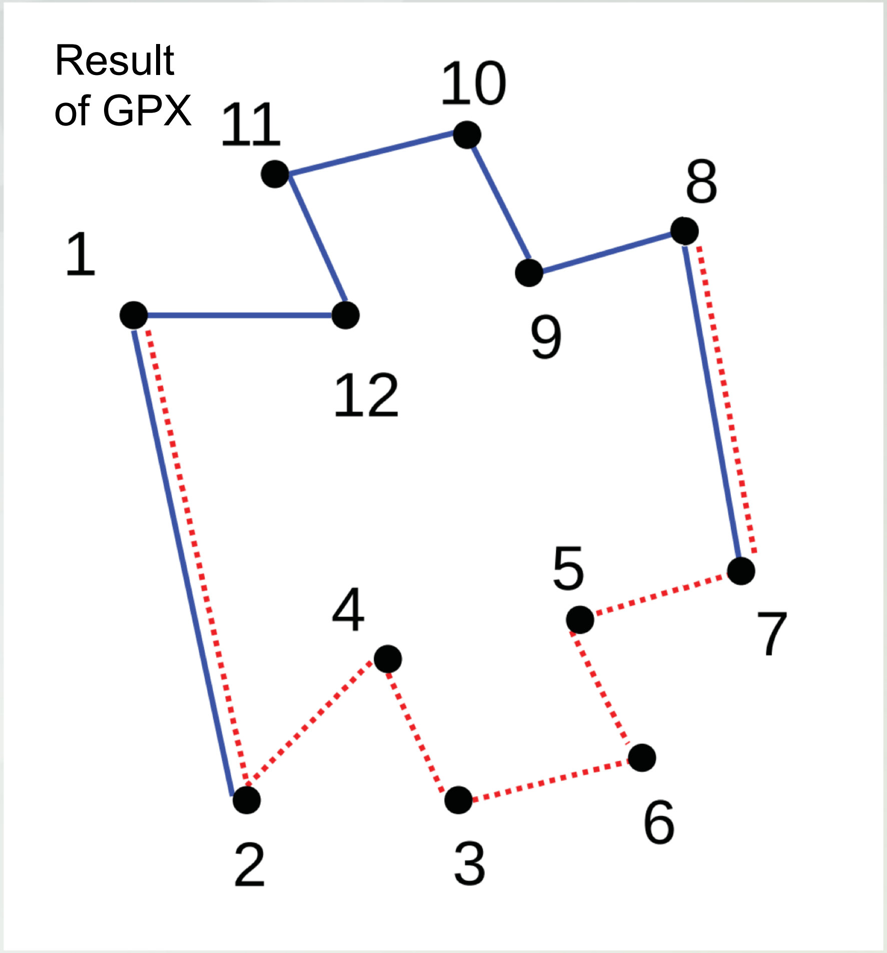
GPX (cont.)

- Thus, each of the candidate components in our example is also a recombining component… follow the traversal of each component (from 7 to 2, then 1 to 8 to 7, for example) by each parent to see that…



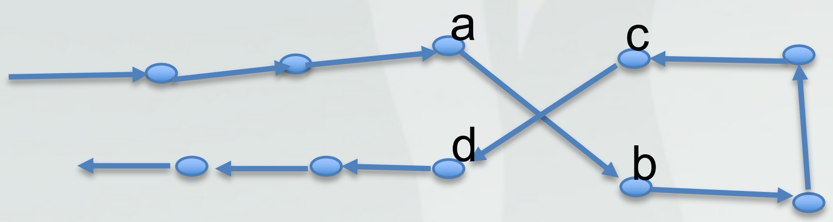
GPX (cont.)

- Now recombine the best partial solutions inside each partition
- Note that we now have the best possible recombination of the two recombining components
- But note that this recombination is *exploitative* – it CANNOT generate any edge that was not present either parent!

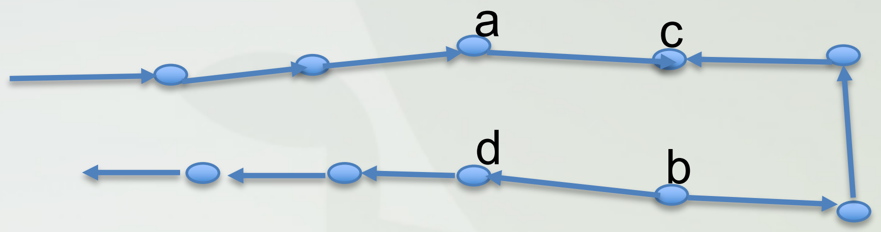


GPX (cont.)

- Thus, GPX is applied in conjunction with other, more exploratory heuristics, in solving the overall problem
- For example, 2-opt local search can find a solution that cannot be improved by swapping any two edges (below, swapping a→b and c→d to become a→c and b→d improves)



Original Tour



2-opt Tour Resulting

- Then, if two recombining components are both 2-opt, then the resulting offspring of GPX will still be 2-opt, and include more cities

GPX – GPX2

- Original GPX has been extended, generating many more recombination opportunities (see the paper)
- Here are some elements improved in GPX2:
  - splitting vertices of degree four to create more connected-subgraphs;
  - identifying recombining components with more than one entry and one exit;
  - merging or "fusing" neighbor candidate components that are not recombining components;
  - scanning one of the two parents in both the forward and reverse directions to identify more recombining components; and
  - merging or "fusing" more complex, potentially nested candidate components with multiple entries and exits

GPX – GPX2

- GPX2, running inside LKH (Lin-Kernighan-Helsgaun) algorithm, is producing world-class results in shorter runtimes than other approaches, on very large problems (~100,000 cities)
- But can it promise to solve to optimality?
- If you want to work on TSP and related combinatorial problems, you should learn this work