Superior Products Through Innovation

Advanced Development Programs

Generalized Weapon Target Assignment Problem

J. Rosenberger, P. Hwang, R. Pallerla, UTA

A. Yucel, R. Wilson, E. Brungardt, LM Aero

presented at

10th ICCRTS, McLean, VA, 14 June 2005

Overview

- Problem Statement
- Branch and Bound Method
- Successive Auction Method
- Computational Experiments
- B & B Extensions
- Summary

Problem - Statement

- **Set of Platforms**
  - ***Resources (type, quantity)***
- **Set of Targets**
- **Assign platforms/resources to targets**
  - ***Assignment: Source {Target/Resource/qty}, Target/Resource/qty},***
  - ***Benefit***
  - ***Target Value***
  - ***Effectiveness (type, quantity)***
- **Goal**
  - ***Maximize overall benefit***

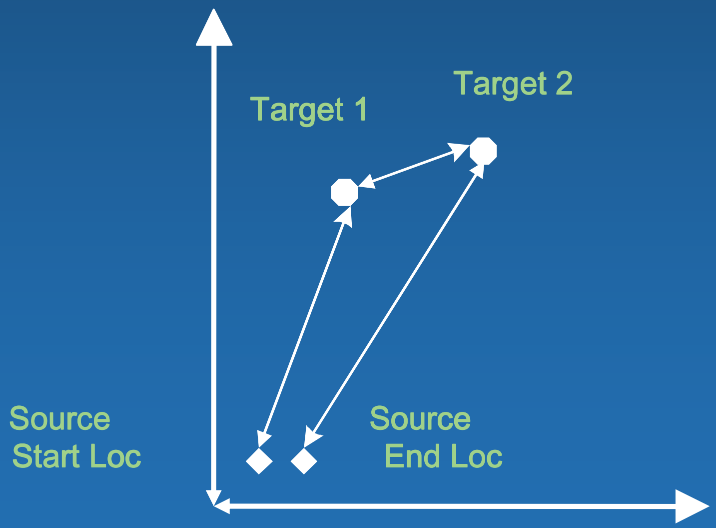
Assignment Problem

- **Basic Assignment Problem**
  - ***m platforms, n targets***
  - ***One target per platform***

- **Generalized Weapon-Target Assignment Problem**
  - ***Multiple target assignments per platform***
  - ***Multiple sources per target***

Enumeration of Assignments

- **Target Drop**
  - *Target/Resource/qty*
  - *Benefit = Target Value * Resource Effectiveness*
- **Target Drop Set**
  - *One or more Target Drops*
  - *Total Benefit = Sum of Target Drop Benefits*
- **Assignment**
  - *Source <–> Target Drop Set*
  - *Benefit = Target Drop Set Benefit * Plength*
  - *Feasible - travel capacity constraint*
  - *Additional considerations*



Problem - Formulation

Formulated as an integer programming problem

$$\max \sum_{j \in J} c_j x_j$$

$$s.t. \sum_{j \in Ss} x_j \leq 1 \quad s = 1, \ldots, m$$

$$\sum_{j \in Tt} x_j \leq 1 \quad t = 1, \ldots, n$$

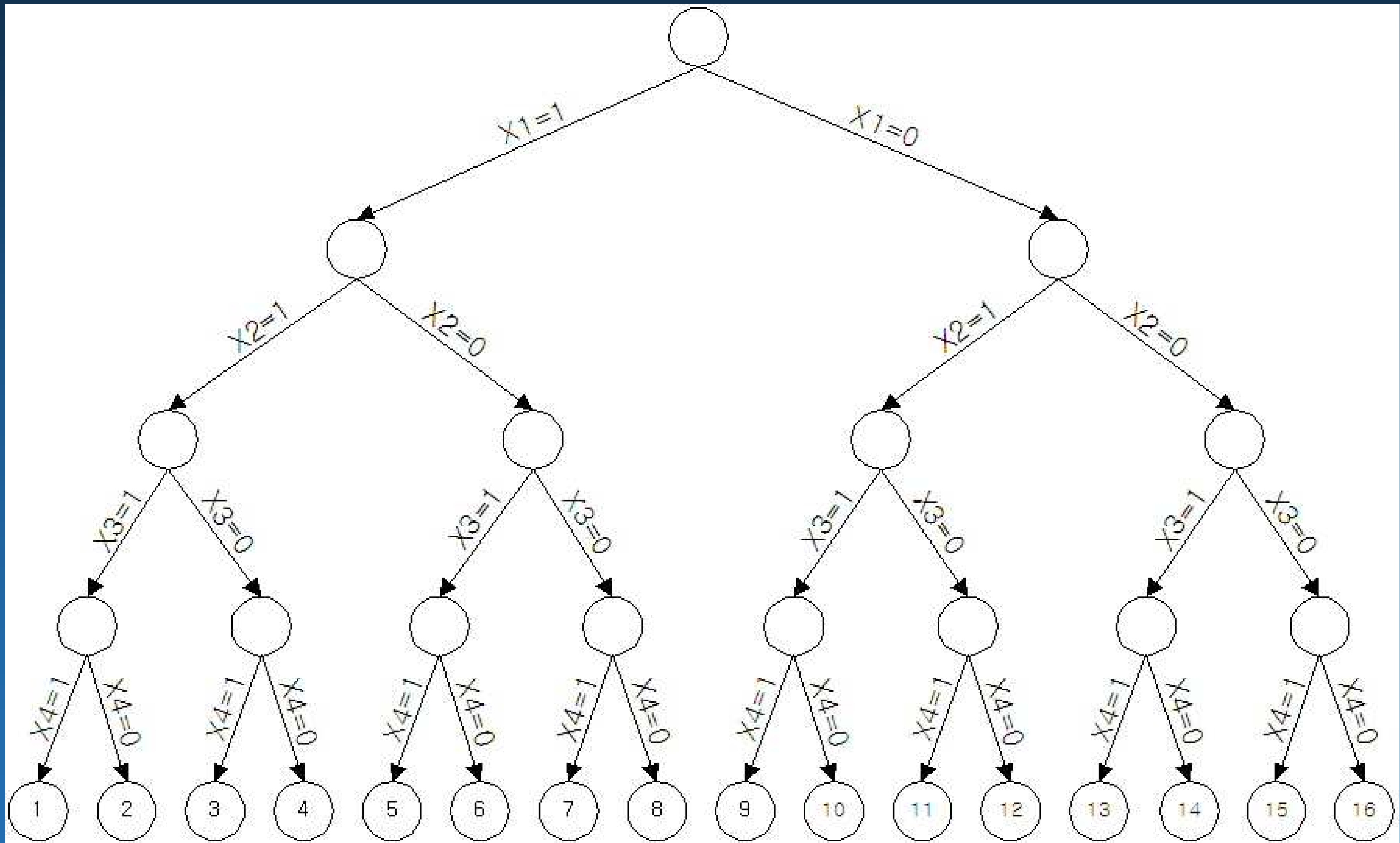
$$x_j = \begin{cases} 1, & \text{if assignment } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

J **is the set of all feasible assignments and** c_j **the benefit from assignment** ***j***

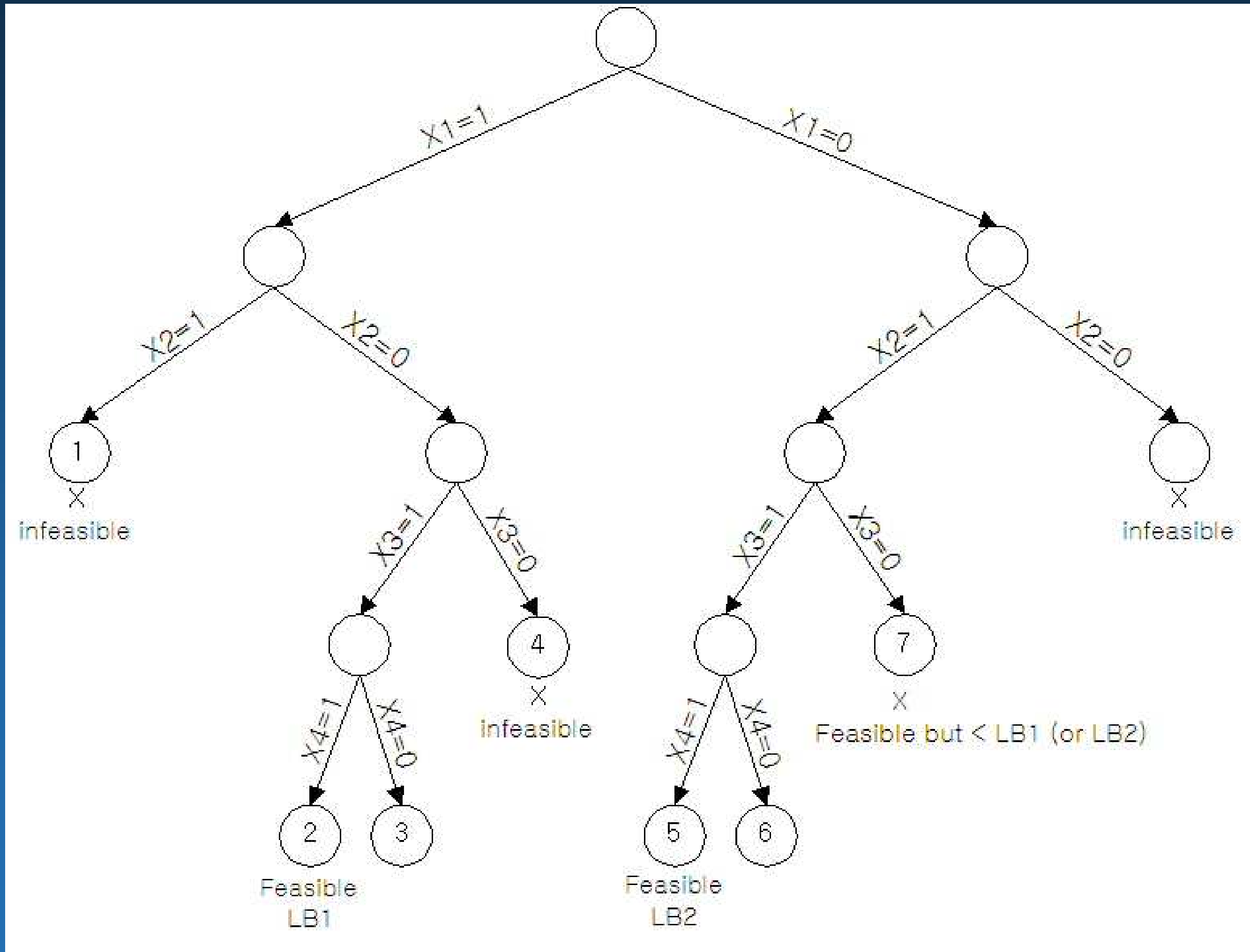
S_s**, s = 1,…,m, is the subset of** ***J*** **to which source s is assigned**

T_t**, t = 1,…,n , is the subset of** ***J*** **which serves target t**

Enumeration Tree Example



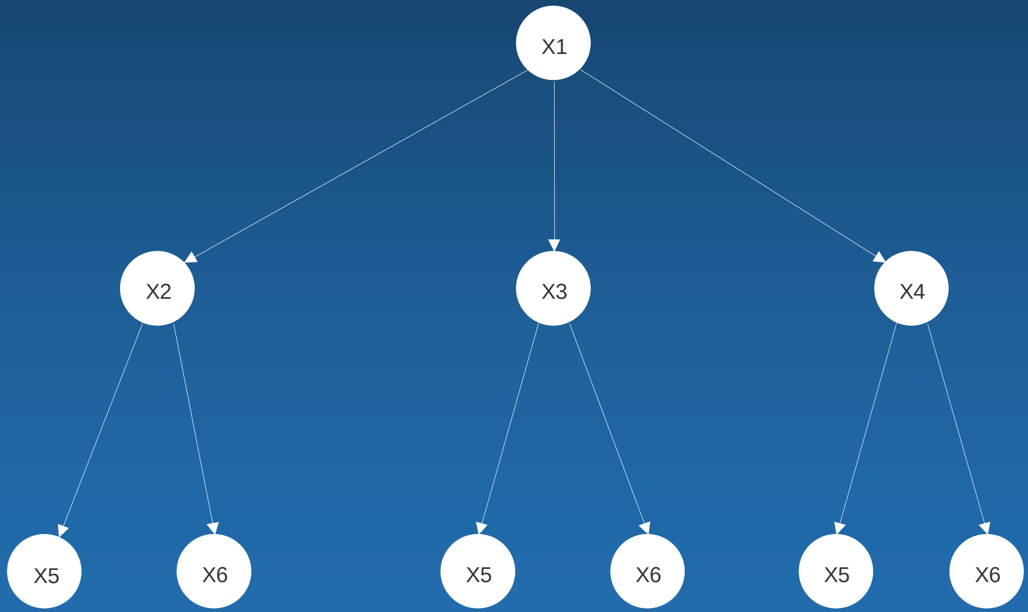
Enumeration Tree Sweep



Branch and Bound - continued

- **Sort Assignments by source in decreasing benefit order**
  - ***Branch on each source***

Source	Assignments
S1	X1
S2	X2 X3 X4
S3	X5 X6



Branch and Bound - continued

- **Incorporate objective tolerance**

  Source conflicts

S1	1	2	3	4	5	6	7
S2	8	9	10	11	12	13	
S3	14	15	16	17	18		

- **If our current optimal solution is at (1,8,15):**

  if

  [total benefit of (1,9,14)] < [total benefit of (1,8,15) – objective tolerance]

  then skip the (1,9,14), (1,9,15), (1,9,16), (1,9,17), (1,9,18) combinations - this will reduce further unnecessary branchings

Branch and Bound - continued

- **Target Conflicts need to be efficiently evaluated**
- **Bitmasks used to characterize targets in an assignment as well as the targets in the current solution**
- **Bitwise AND of current solution target representation (through source levels n) with the target representation for an assignment for source n+1**
- **Single integer for up to 32 targets, use arrays if > 32**

Branch and Bound - continued

- **Will find an assignment with optimal benefit**
- **Large number of enumerations – can be costly**
- **Number of combinations may prove intractable for large m, n**

Sequential Method

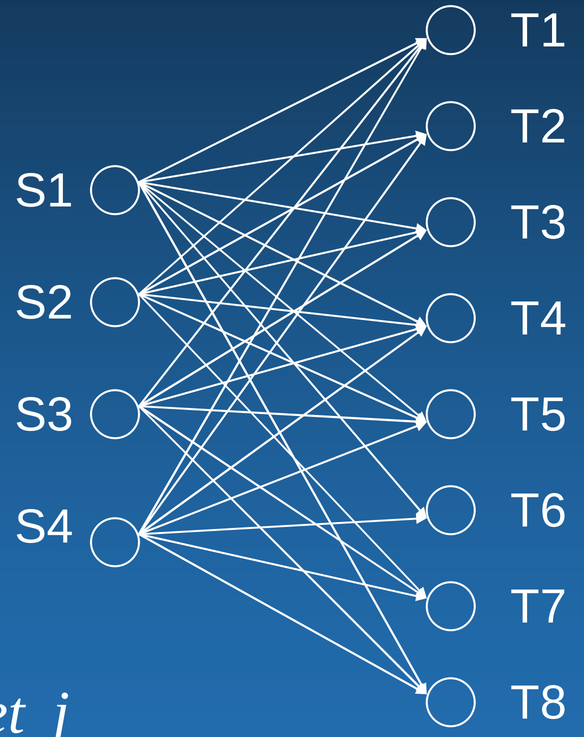
Given m sources and n targets, we set up a directed bipartite graph (G) of sources and targets, and look for the solution of the asymmetric assignment problem

$$\max \sum_{(i,j)\in G} a_{ij} x_{ij}$$

$$s.t. \quad \sum_{\{j|(i,j)\in G\}} x_{ij} \leq 1 \quad i = 1,\ldots,m$$

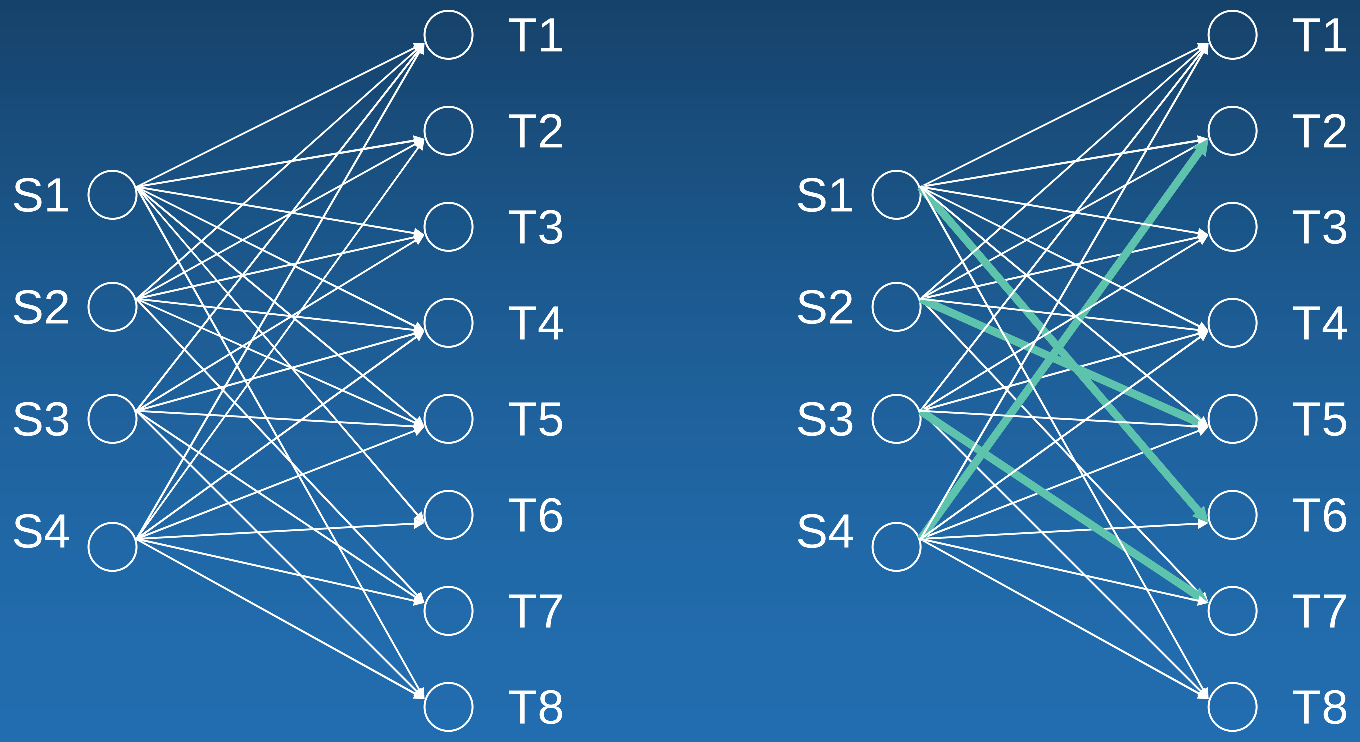
$$\sum_{\{i|(i,j)\in G\}} x_{ij} \leq 1 \quad j = 1,\ldots,n$$

$$x_{ij} = \begin{cases} 1, & if \ \ source\ i\ assigned\ to\ target\ \ j \\ 0, & otherwise \end{cases}$$



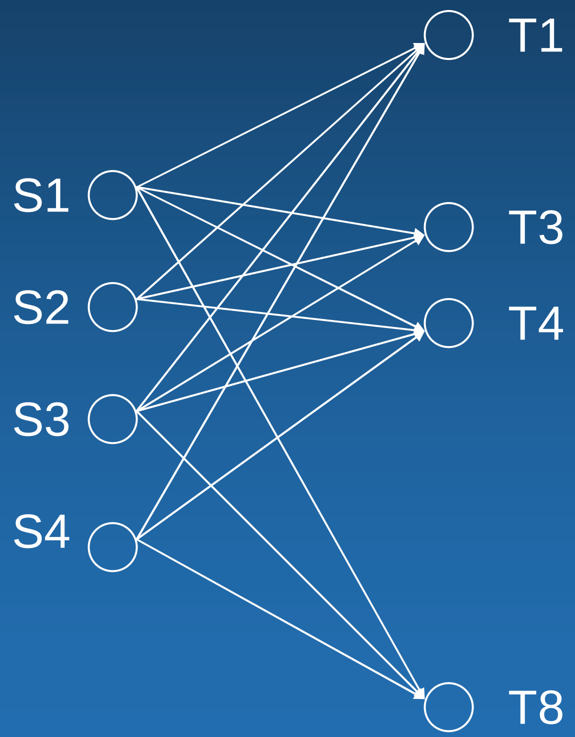
Sequential Method – Auction algorithm

- **Use Bertsekas auction algorithm – multiple passes for m < n**
  - ***First pass***



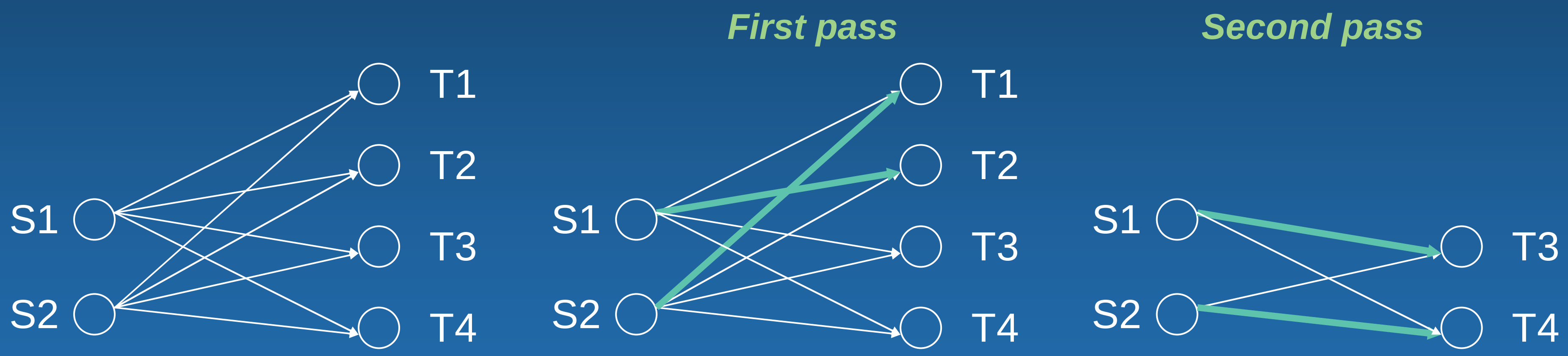
Sequential Method – Auction algorithm

- **Successive passes**
  - *Regenerate graph with remaining resources and targets*



Sequential Method

- Fast, less memory, fewer enumerations
- Greedy approach – may yield less than optimal solution
  - *S1 with Ax2, S2 with Bx2*
  - *T1 (100 ; Ax1, 1.0 or Bx1,1.0), T2 (100 ; Ax1, 1.0 or Bx1,1.0)*
  - *T3 (100 ; Ax1, 0.5 or Bx1,1.0), T4 (100 ; Ax1, 0.5 or Bx1, 1.0)*



Sequential method solution: S1 – T2 – T3, S2 – T1 – T4, Total benefit 350

Branch & bound method solution: S1 – T2 – T1, S2 – T3 – T4, Total benefit 400

Computational Experiments

- Created Monte carlo simulation code for generating different set of sources, targets and their resource requirements, randomly
- Base Parameter Set

Max Targets	2
Max Sources	1
Travel Capacity	25 units
Objective Tolerance	0
Max Time	10
Max Nodes	5000000

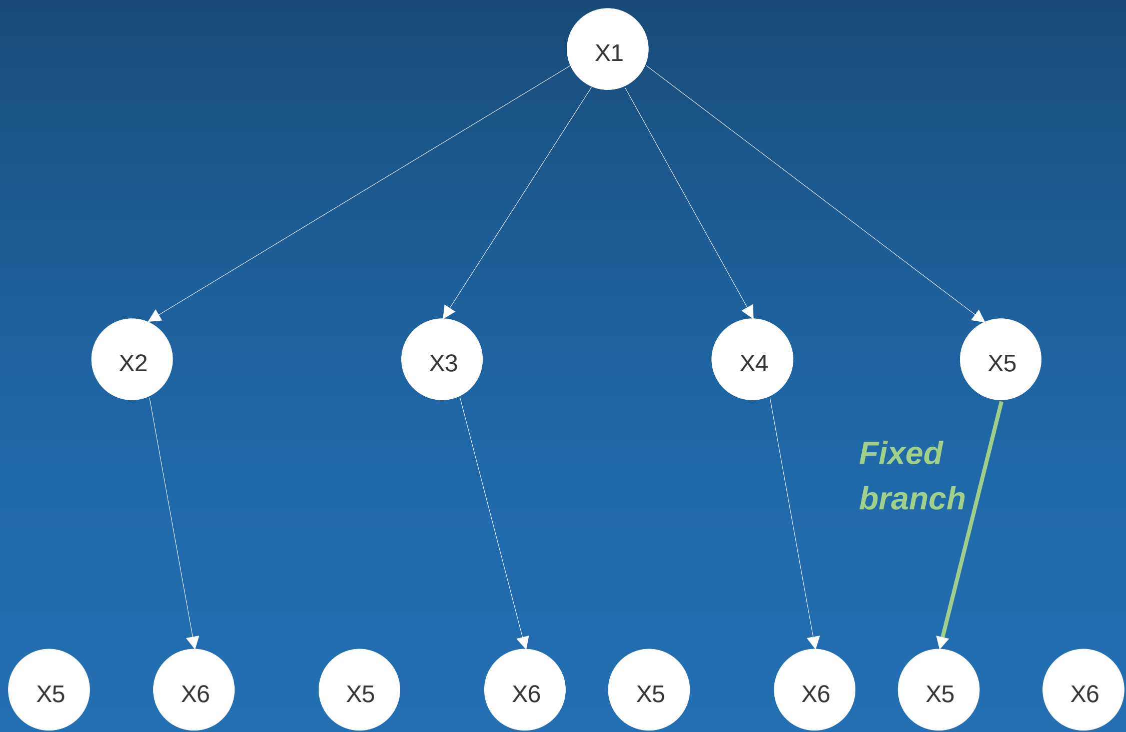
- By modifying one variable in the base every time, we generated different parameter sets
- In each parameter set, we generated 50 instances. Average benefit of those 50 solutions benefit by the Branch and Bound algorithm is compared with the average of the 50 solutions by sequential method
- Sequential method solutions within 1% - 5% of Branch & Bound solutions

Branch & Bound Method – Multiple Targets

- **Allow more than one source per target**
  - ***e.g. Two platforms, each delivering half the required resource for a given effectiveness***

- **Example**

Source	Assignments
S1	X1
S2	X2 X3 X4 X5
S3	X5 X6



Conclusions

- **Investigated suitability of successive auction algorithm for multi-target assignments**
  - ***Branch and Bound serving as benchmark***
  - ***Fast, efficient, fewer enumerations: $O(nm)$ vs. $O(nm^2)$ for 2 targets per source***
  - ***Successive auctions consistently found multiple assignments close to optimal***

- **Gained insight into performance of B&B approach**
  - ***First feasible solution also close to optimal value***
  - ***Differences may be deemed operationally insignificant***
  - ***Can thus provide quick solutions to complicated multi-source assignment problems***

- **Questions?**