AUTOMATED PLANNING AND SCHEDULING USING CALCULUS OF VARIATIONS IN DISCRETE SPACE

Yixin Chen and Benjamin W. Wah

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
http://manip.crhc.uiuc.edu

Work supported by NASA Contract NCC 2-1230

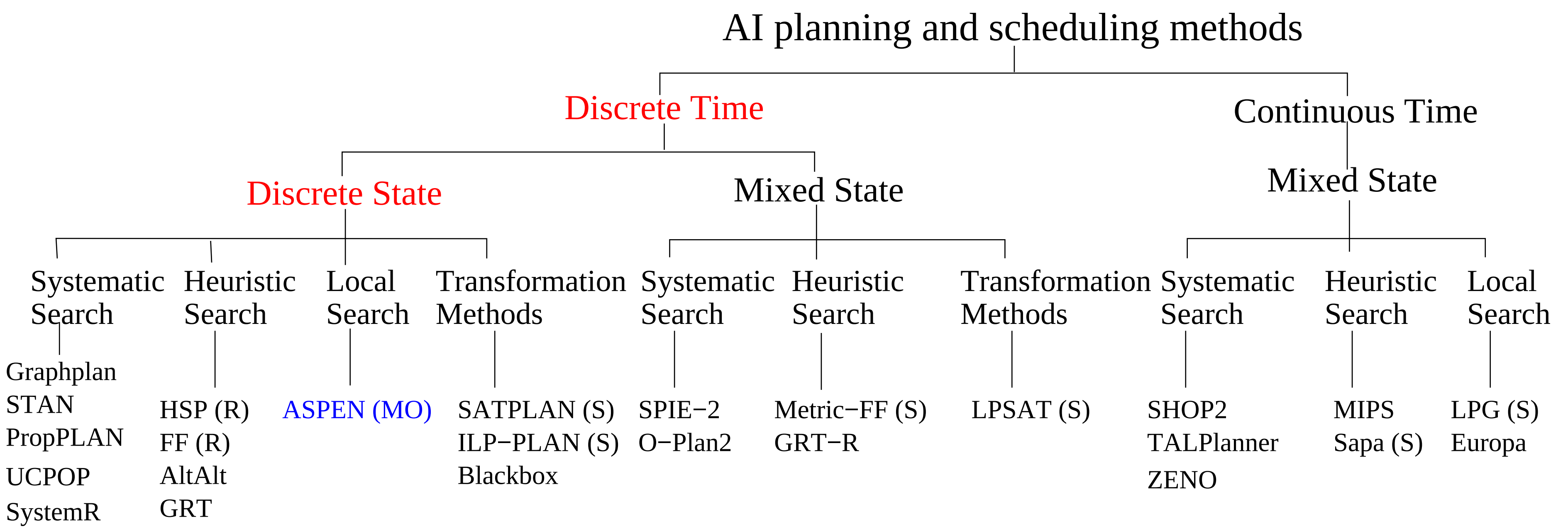
June 11, 2003

Outline

- Introduction
  - Research problem addressed
  - Limitations of dynamic programming and penalty methods
- Theory of Lagrange multipliers for discrete constrained optimization
  - Necessary and sufficient saddle-point condition
  - Iterative implementation
- Partitioning of variable space
  - Distributed necessary saddle-point condition
  - Distributed Iterative implementation
- Experimental results on ASPEN
- Conclusions

INTRODUCTION

A Classification of Existing Approaches in Planning

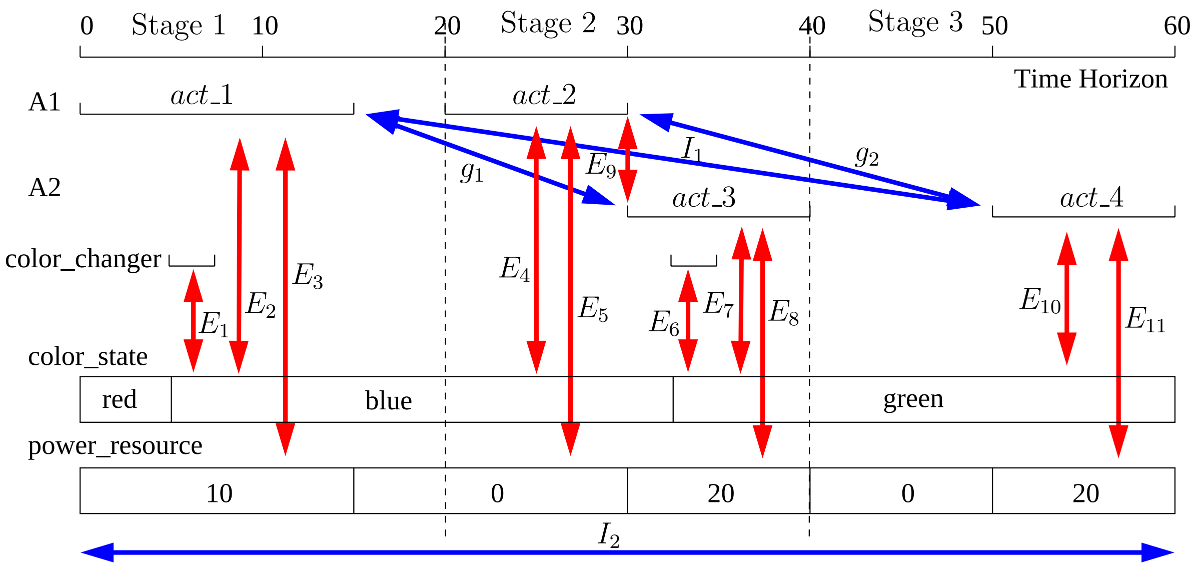


ASPEN

- Discrete time horizon and discrete space
- Discrete temporal and metric constraints
- Multiple preferences combined in a weighted sum
- Greedy local optimization of objective and repair-based constraint satisfaction

Toy Example solved by ASPEN

```
model toy {HORIZON_START = 1998-1/00:00:00; horizon_duration = 60s; time_scale = second;};
parameter string color {domain = ("red", "blue", "green");};
State_variable color_sv {states = ("red", "blue", "green"); default_state = "red";};
Resource power {type = non_depletable; capacity = 25; min_value = 0;};
Activity color_changer {color c; duration = 1; reservations = color_sv change_to c;};
Activity A1 {duration = [10,20]; constraints = ends_before start of A2 by [0,30];
       reservations = power use 10, color_sv must_be "green";};
Activity A2 {duration = 10; reservations = power use 20, color_sv must_be "blue";};
// initial schedule
A1 act_1 {start_time = 0; duration = 15;};   act_2 { start_time = 20; duration = 10;};
A2 act_3 {start_time = 30; duration = 10;};  act_4 { start_time = 50; duration = 10;};
```

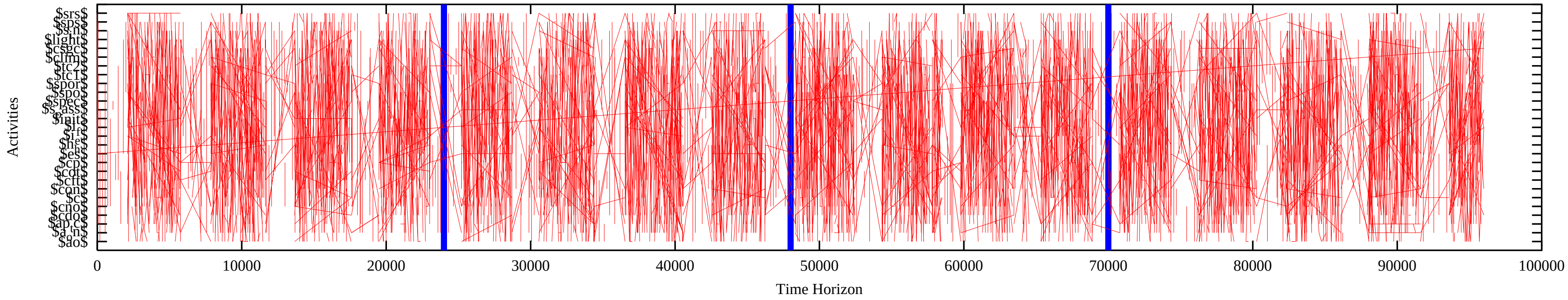


Local constraints ($E_1, \cdots, E_{11}$):
- color_state constraints for *act*_1, act_2, *act*_3,
- power_resource constraints for *act*_1 thru *act*_4;
- color_state transition constraints relating color_changer and color_state
- *act*_2 ends_before start of *act*_3 by [0,30].

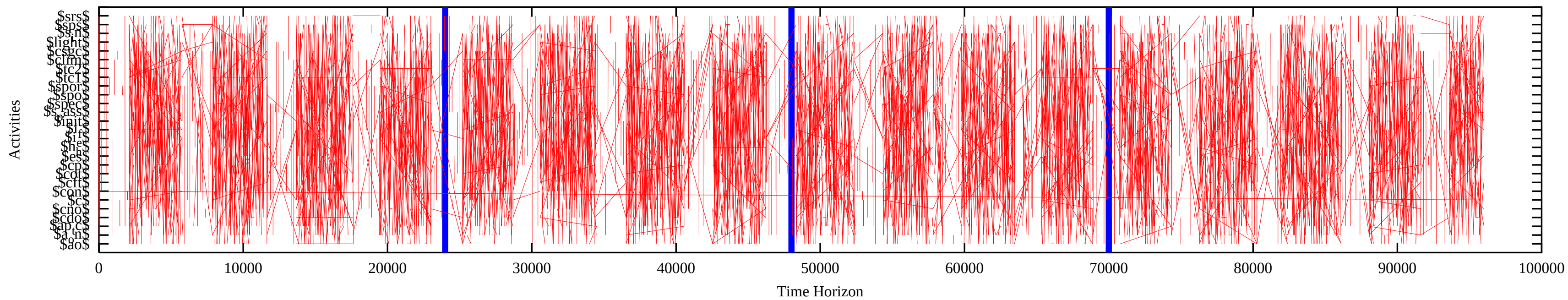
General constraints:
- *act*_1 ends_before start of *act*_3 by [0,30] (g_1);
- *act*_2 ends_before start of *act*_4 by [0,30] (g_2).
- *act*_1 ends_before start of *act*_4 by [0,30] (I_1);
- power_resource always less than capacity of power_resource (I_2).

Solving CX1-PREF with 16 Orbits (with 3,687 constraints) by ASPEN

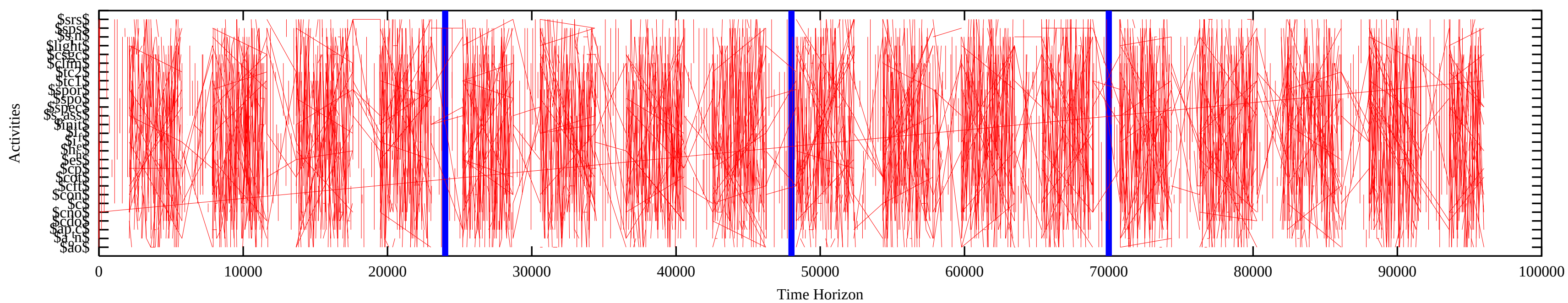
Initially



1,000 Iter.



1,0000 Iter.

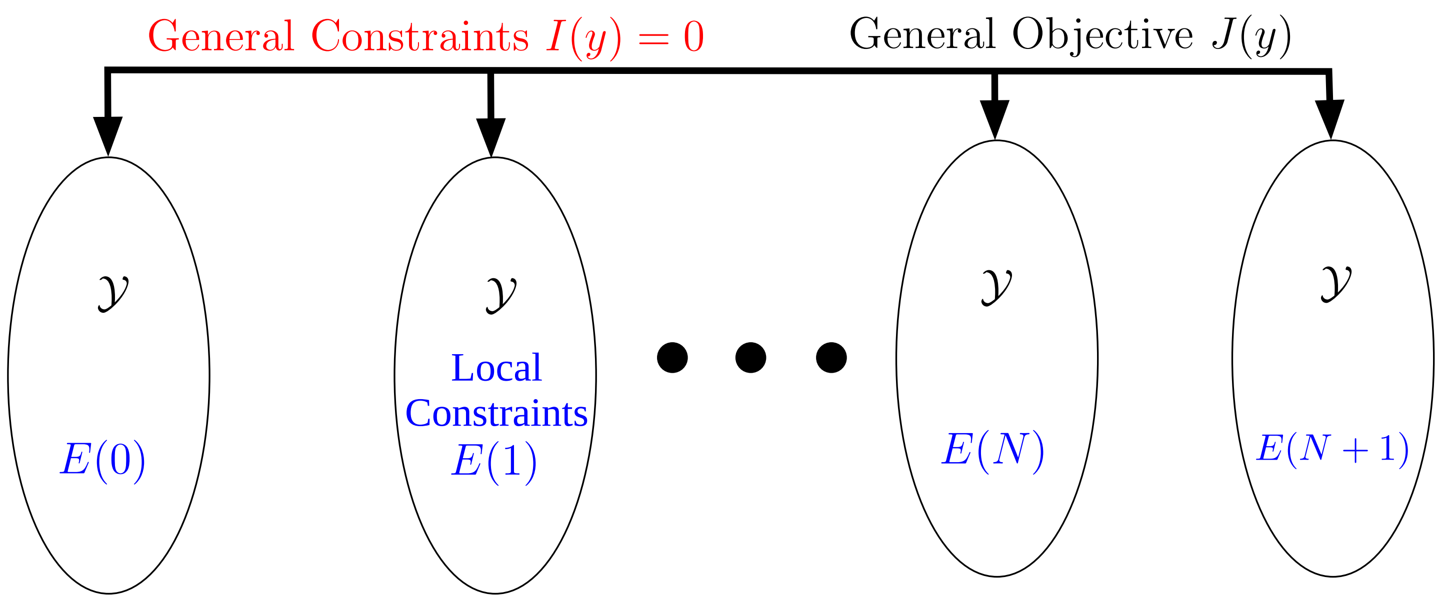


Research Problem Addressed

Partition (discrete-space and discrete-time) temporal planning problems and develop methods for resolving general constraints across partitions

- Partitioned problems have lower time and space complexity
- Overall problem can be solved better and more efficiently

Mathematical Formulation

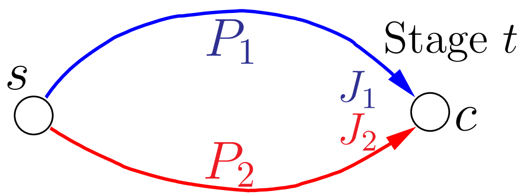


$$\begin{aligned} \min_{y} \quad & J(y) \\ \text{subject to} \quad & E(j, y(j)) = 0, \quad j = 0, 1, \cdots, N+1 \\ & I(y) = 0 \end{aligned}$$

where $y(j)$ is defined in *discrete* space $\mathcal{Y}$ of stage j,
E, I and J are *not* necessarily continuous and differentiable

Dynamic Programming Cannot Be Applied

- Path dominance on multi-stage search with local constraints
  - Principle of Optimality applied on feasible state c



If c lies on the optimal path between s and d and $J_2 \leq J_1 \quad \Longrightarrow \quad P_2 \rightarrow P_1$

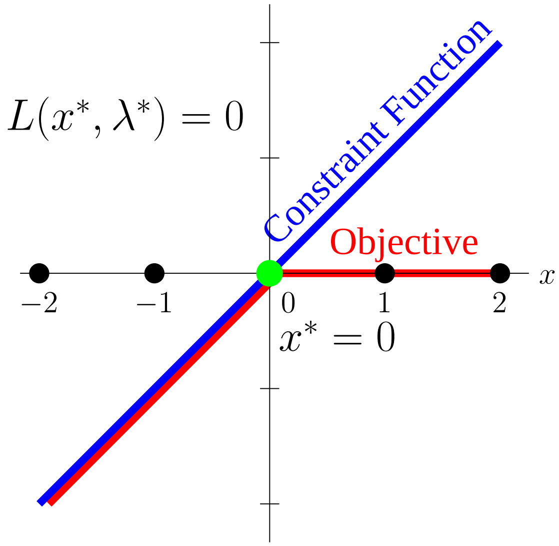
  - Polynomial worst-case complexity: $O\left(N|\mathcal{Y}|^2\right)$
- Path dominance is not applicable when there are general constraints
  - A dominating path may become infeasible due to general constraints
  - Exponential search space: $O\left(|\mathcal{Y}|^{N+2}\right)$

Penalty-Based Methods Do Not Always Work

Penalty-based methods

- By choosing suitable penalties in a penalty function, a local minimum of the penalty function corresponds to a feasible local minimum of the objective

Counter-example



$$\min_{x \in \{-2,-1,0,1,2\}} f(x) = \begin{cases} 0 & x \geq 0 \\ x & x < 0 \end{cases}$$

subject to $x = 0$

Penalty formulation

- $L(x, \lambda) = f(x) + \lambda\, x$
- Hypothesize $L(x, \lambda^*) \geq L(x^*, \lambda^*) = 0$

No λ^* exists when solving

$$\begin{cases} L(-1, \lambda^*) = -1 - \lambda^* \geq L(0, \lambda^*) = 0 \\ L(1, \lambda^*) = 0 + \lambda^* \geq L(0, \lambda^*) = 0 \end{cases}$$

$$\Longrightarrow \begin{cases} \lambda^* \leq -1 \\ \lambda^* \geq 0 \end{cases}$$

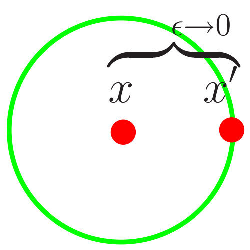
THEORY OF LAGRANGE MULTIPLIERS FOR DISCRETE CONSTRAINED OPTIMIZATION

Neighborhood $\mathcal{N}(x)$ of Point x

Continuous Space: $\mathcal{N}_{cn}(x)$

x is a vector of continuous variables

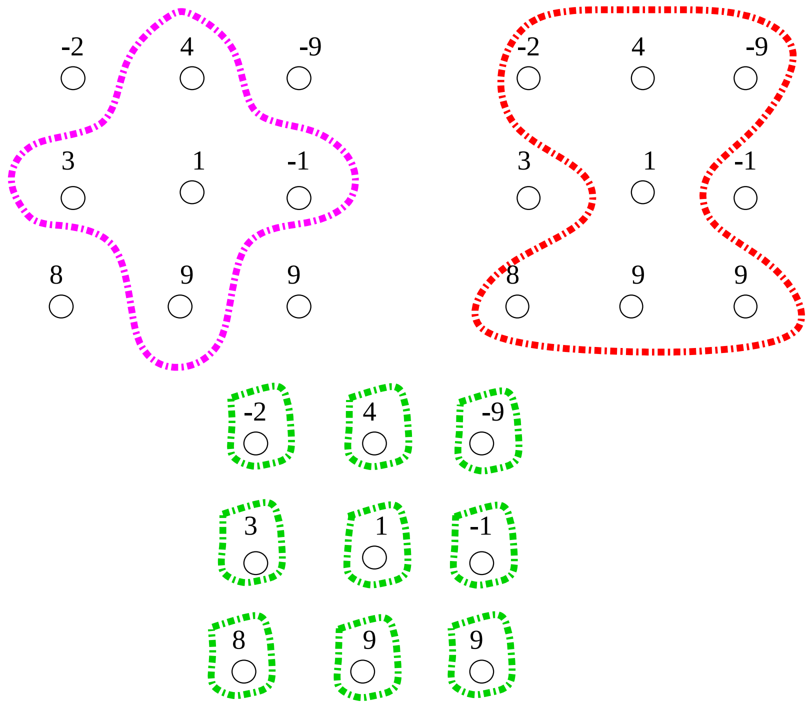
Neighborhood defined by open sphere



Discrete Space: $\mathcal{N}_{dn}(x)$

x is a vector of discrete variables

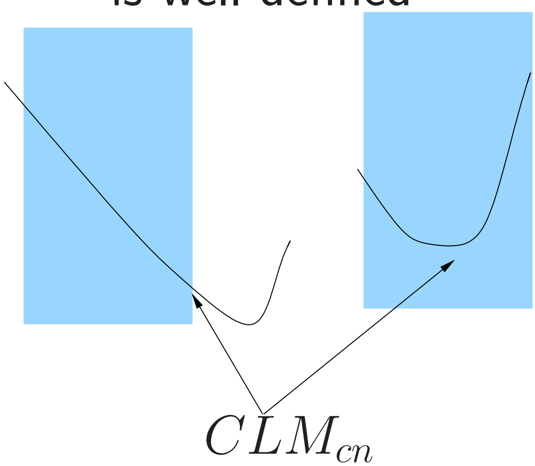
User defined neighborhood



Constrained Local Minimum (CLM)

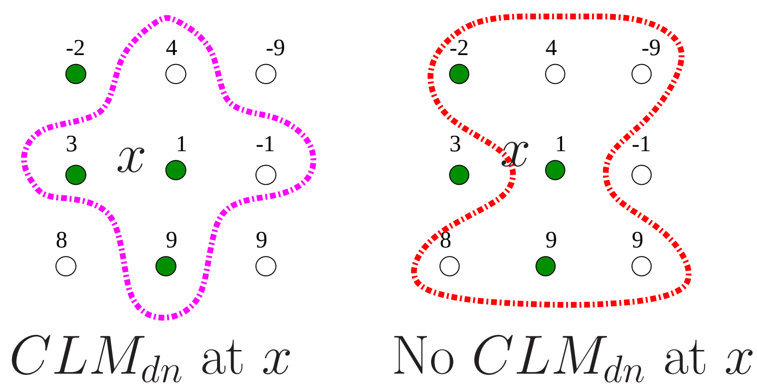
Continuous Space: CLM_{cn}

- Feasible local minimum when compared to feasible points inside an open sphere
- Whether point x is a CLM_{cn} is well defined



Discrete Space: CLM_{dn}

- Feasible local minimum with respect to neighboring feasible points
- Whether point x is a CLM_{dn} depends on $\mathcal{N}_{dn}(x)$



Lagrangian Formulation of Discrete Optimization Problem

- Let H be a transformation function where $H(x) \geq 0$ and $H(x) = 0$ iff $x = 0$

$$L_{dn}(y, \gamma, \mu) = J(y) + \sum_{t=0}^{N+1} \gamma^T(t) H\left(E(t, y(t))\right) + \mu^T H\left(I(y)\right)$$

Lagrangian Formulation of Discrete Optimization Problem

- Let H be a transformation function where $H(x) \geq 0$ and $H(x) = 0$ iff $x = 0$

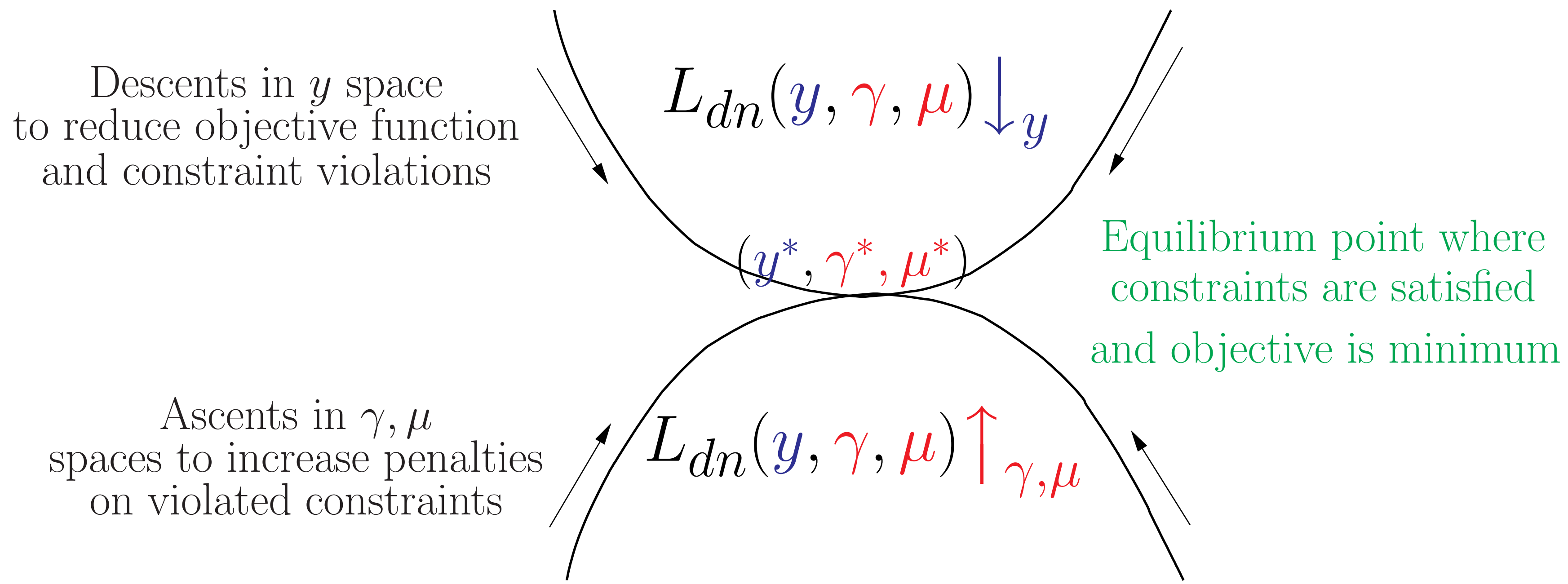
$$L_{dn}(y, \gamma, \mu) = J(y) + \sum_{t=0}^{N+1} \gamma^T(t) \cdot \Big| E(t, y(t)) \Big| + \mu^T \cdot \Big| I(y) \Big|$$

- Necessary and sufficient saddle-point condition
  - y^* is a CLM_{dn} iff (y^*, γ^*, μ^*) is a discrete-neighborhood saddle point (SP_{dn})

$$L_{dn}(y^*, \gamma, \mu) \leq L_{dn}(y^*, \gamma^*, \mu^*) \leq L_{dn}(y, \gamma^*, \mu^*)$$

- (y^*, γ^*, μ^*) is at
  - Local minimum of L_{dn} with respect to y
  - Local maximum of L_{dn} with respect to γ and μ
- Condition is true for $\gamma^{**} \geq \gamma^*$ and $\mu^{**} \geq \mu^*$

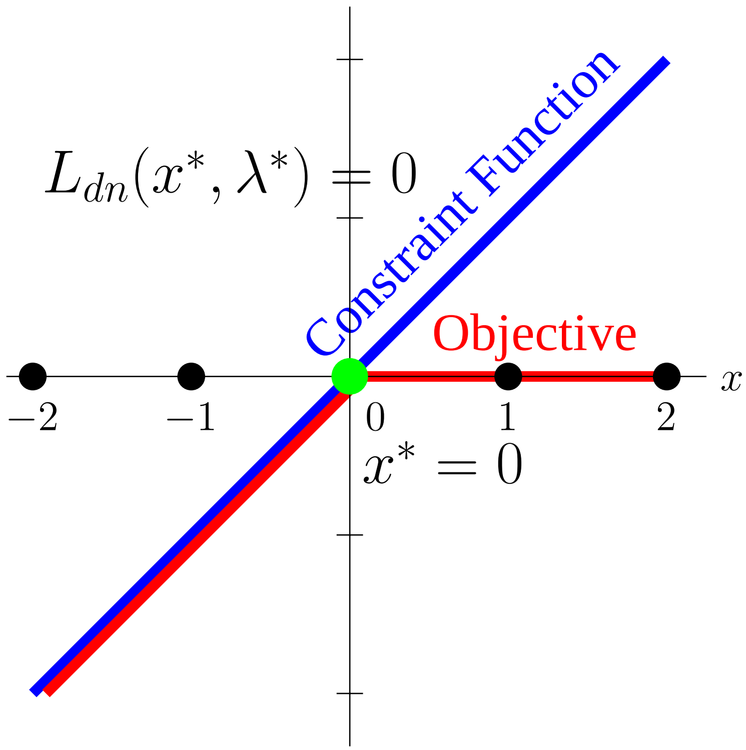
Intuitive Meaning Behind Saddle Points



Although γ^* and μ^* always exists,

- Their search in discrete space may be very time consuming
- The search of $\gamma^{**} \geq \gamma^*$ and $\mu^{**} \geq \mu^*$ is much easier

Continuing from the Previous Example



$$\min_{x \in \{-2,-1,0,1,2\}} f(x) = \begin{cases} 0 & x \geq 0 \\ x & x < 0 \end{cases}$$

subject to $x = 0$

Lagrangian formulation

- $L_{dn}(x, \lambda) = f(x) + \lambda\,|x|$
- Find λ^* such that $L_{dn}(x, \lambda^*) \geq L_{dn}(x^*, \lambda^*)$

Solving

$$\begin{cases} L_{dn}(-1, \lambda^*) = -1 + \lambda^* \geq L_{dn}(0, \lambda^*) = 0 \\ L_{dn}(1, \lambda^*) = 0 + \lambda^* \geq L_{dn}(0, \lambda^*) = 0 \end{cases}$$

leads to $\lambda^* \geq 1$

Pick $\lambda^* = 1$
Saddle-point condition applies for $\lambda^{**} \geq \lambda^*$

Iterative Implementation

Algorithm needs to look for $\gamma^{**} \geq \gamma^*$ and $\mu^{**} \geq \mu^*$

$L_{dn}(y, \gamma, \mu) \uparrow_{\gamma,\mu}$ to find γ^{**}, μ^{**}

Outer Loop

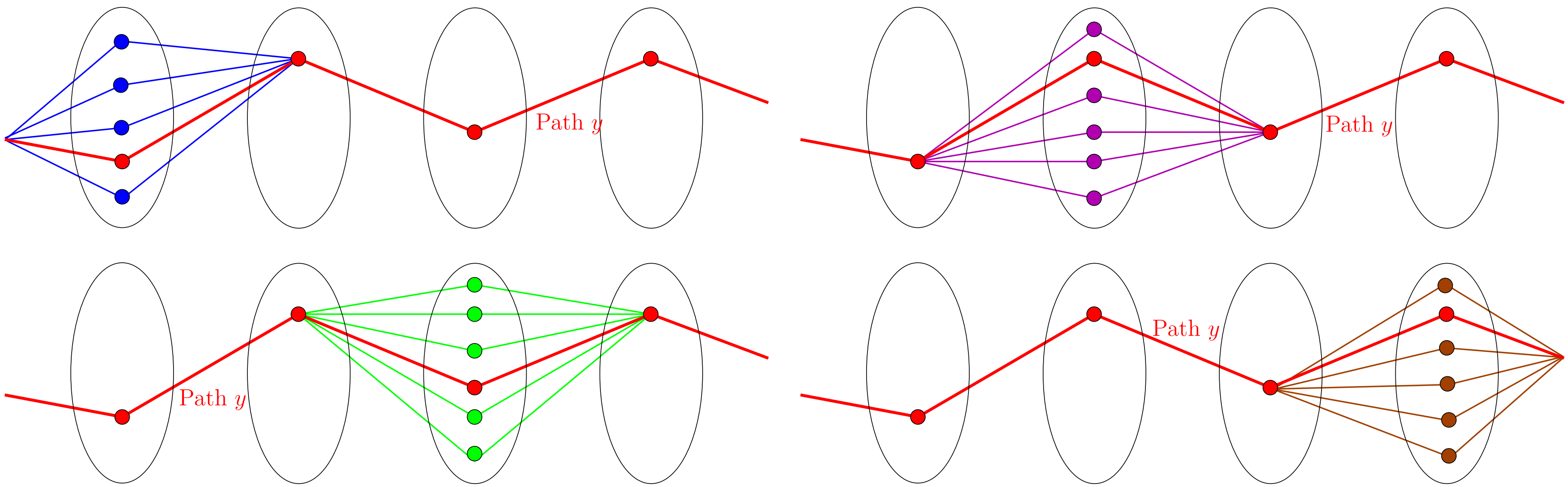
$L_{dn}(y, \gamma, \mu) \downarrow_{y}$ to find y^*

Inner Loop

PARTITIONING OF VARIABLE SPACE

Neighborhood in Partitioned Variable Space

$\mathcal{N}_b(y)$ (discrete neighborhood of path $y = (y(0), \ldots, y(N+1))^T$) is the union of discrete neighborhoods in each stage, while keeping the path fixed in other stages



Path y is a constrained local minimum in discrete space (CLM_{dn}) iff

- y is feasible
- No feasible path in $\mathcal{N}_b(y)$ has better objective value than $J(y)$

Decomposition of Lagrangian Function into Stages

Decompose Lagrangian function

$$L_{dn}(y, \gamma, \mu) = J(y) + \sum_{t=0}^{N+1} \gamma^T(t) \cdot \big|E(t, y(t))\big| + \mu^T \cdot \big|I(y)\big|$$

into distributed Lagrangian function for stage t, $t = 0, \cdots, N+1$,

$$\Gamma_{dn}(t, y, \gamma(t), \mu) \;=\; J(y) + \gamma(t) \cdot \big|E(t, y(t))\big| + \mu \cdot \big|I(y)\big|$$

Distributed Necessary Conditions for CLM_{dn}

- Path y is a CLM_{dn} when it satisfies
  - Distributed necessary discrete-neighborhood saddle-point conditions

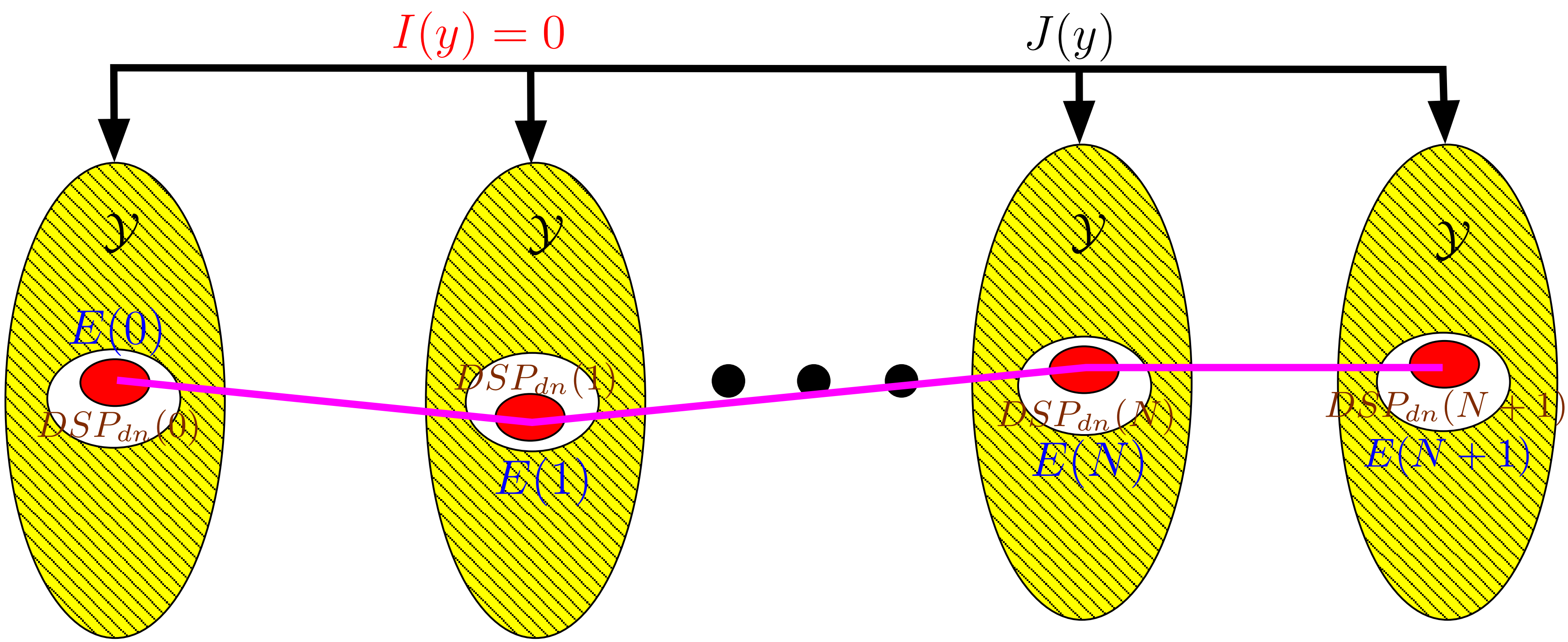
    $(DSP_{dn}(t))$ for all $t = 0, 1, \cdots, N+1$ and fixed μ^*

$$\Gamma_{dn}(t, y^*, \gamma(t)', \mu^*) \leq \Gamma_{dn}(t, y^*, \gamma(t)^*, \mu^*) \leq \Gamma_{dn}(t, y', \gamma(t)^*, \mu^*)$$

$$\text{for all } y' = (y(0), \ldots, y(t-1), y(t)', y(t+1), \ldots, y(N+1)) \in \mathcal{N}_b^{(t)}(y^*)$$

  - Condition is also true for all $\gamma(t)^{**} \geq \gamma(t)^*$

Reduced Search Space for Finding Feasible/Optimal Paths



Significant reduction in complexity

Iterative Implementation

$L_{dn}(y, \gamma, \mu)\uparrow_{\mu}$ to find μ^{**}

$\Gamma_{dn}(0, y, \gamma, \mu)\uparrow_{\gamma(0)}$ to find $\gamma^{**}(0)$

$\Gamma_{dn}(0, y, \gamma, \mu)\downarrow_{y(0)}$

• • •

$\Gamma_{dn}(N+1, y, \gamma, \mu)\uparrow_{\gamma(N+1)}$ to find $\gamma^{**}(N+1)$

$\Gamma_{dn}(N+1, y, \gamma, \mu)\downarrow_{y(N+1)}$

Observation:

- Based on the neighborhood of partitioned variable space, the combined local minimum of Γ_{dn} in all subspaces is the local minimum of L_{dn}

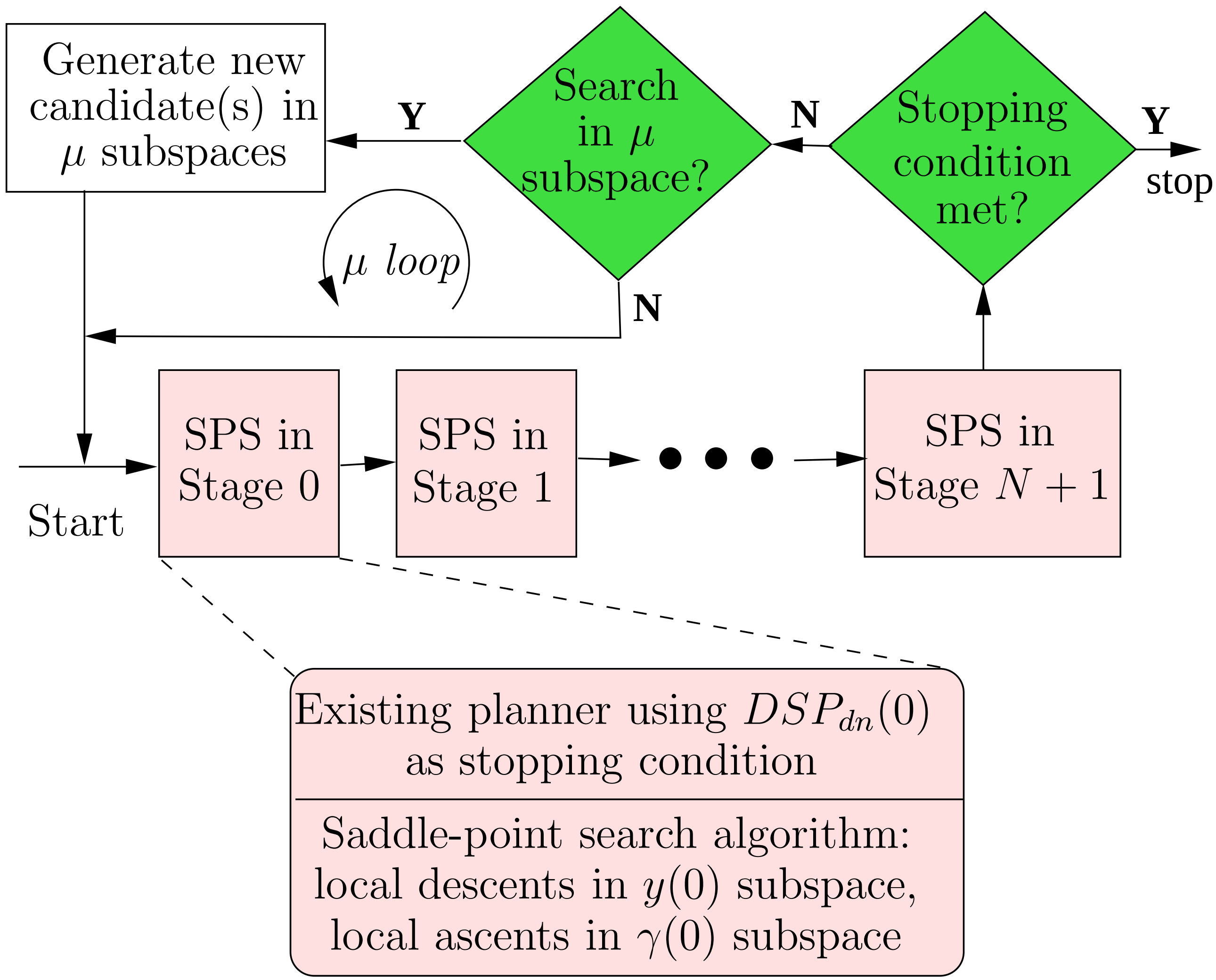
Equivalent Search:

$L_{dn}(y, \gamma, \mu)\uparrow_{\mu}$ to find μ^{**}

$L_{dn}(y, \gamma, \mu)\uparrow_{\gamma}$ to find γ^{**}

$L_{dn}(y, \gamma, \mu)\downarrow_{y}$ to find y^{*}

Heuristic Search Procedure for Finding Local Optimal Paths



EXPERIMENTAL RESULTS ON ASPEN

Approach

1. Dynamically partition problem into $N = 100$ stages with a balanced number of conflicts in each
2. In each stage, perform a certain number of descents and ascents
   - Choose probabilistically from repair actions and optimization actions, and select a random feasible choice at each choice point to create an action
   - Apply action using ASPEN
   - Evaluate augmented distributed Lagrangian function of stage t:

$$\Gamma_{dn}(t) = -w_s \cdot \text{Score} + \sum_{i \in E(t,y(t))} \gamma_i \cdot c_i + \sum_{i \in E(t,y(t))} \frac{c_i^2}{2} + \sum_{j \in I(y)} \mu_j \cdot d_j$$

Score $\in [0, 1]$: preference score of schedule
$w_s = 100$: weight of Score
$c_i = 1$: non-negative value on the degree of violation of local conflict i
$d_j = 1$: non-negative value on the degree of violation of global conflict j

- Accept schedule according to Metropolis probability controlled by a geometrically decreasing T (initial $T = 1000$; cooling rate $= 0.8$)
- For each descent, perform an ascent in γ_i space on violated local conflicts

$$\gamma_i \longleftarrow \gamma_i + \alpha_i \cdot c_i \qquad \text{where } \alpha_i = 0.1$$

3. After iterating over all stages, perform an ascent in μ_j space on violated global conflicts

$$\mu_j \longleftarrow \mu_j + \alpha_j \cdot d_j \qquad \text{where } \alpha_j = 0.1$$

4. If maximum number of iterations is not exceeded, go to (1)

Without UNDO in ASPEN

- Apply selected action to current schedule in a child process
- Repeat the same action in the parent process if action is accepted;
- Discard the result of the child process
- The number of forks is OS dependent and set to 24,000 in our experiments

Benchmark CX1-PREF

- Citizen Explorer-I satellite design and operation planning benchmark
  - Multiple competing preferences to be optimized
  - Problem generator to generate different problem instances

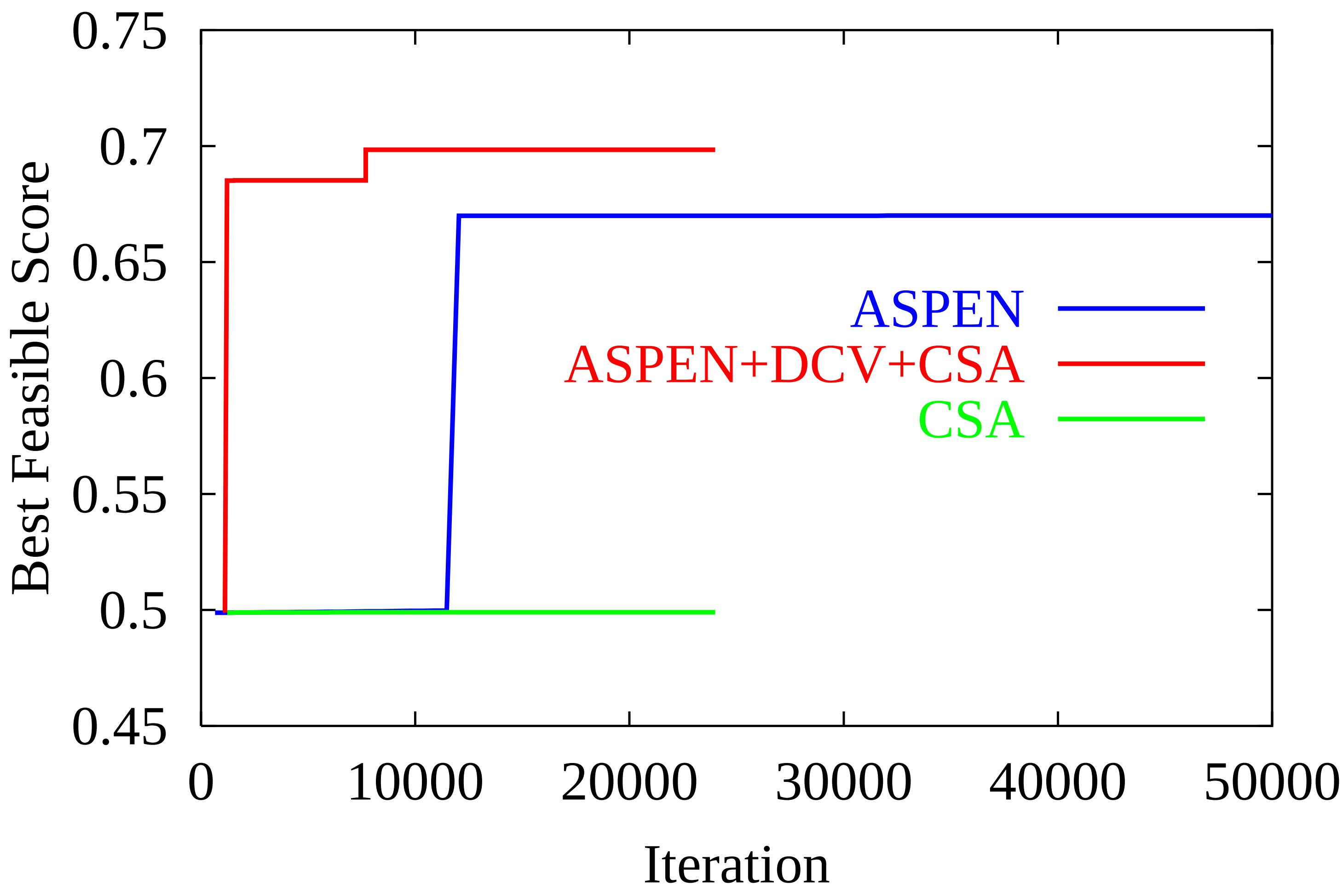
  $perl \quad probgen.pl \quad < random\ seed > \quad < number\ of\ orbits >$

- In default ASPEN search, repeat the following steps

  a) Find feasible schedule using *repair*

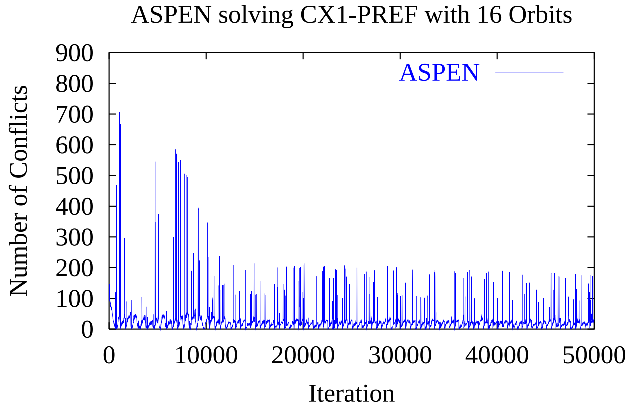
  b) Optimize score using *optimize* (default 200 iterations)

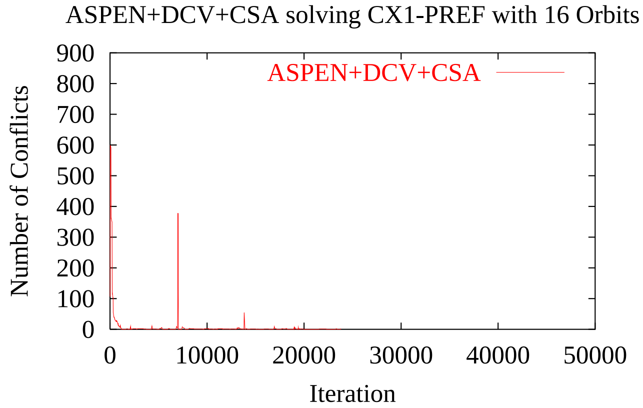
Best Feasible Solution on CX1-PREF with 16 Orbits



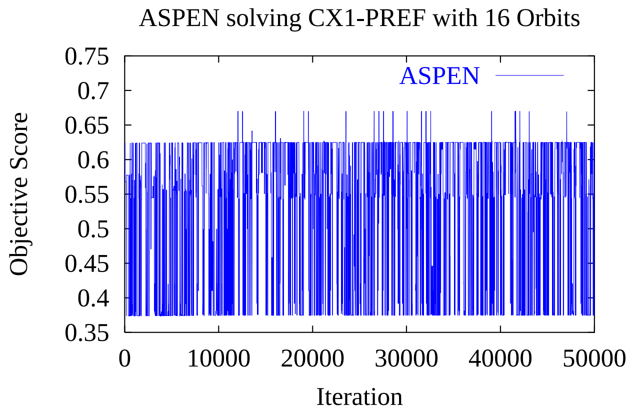
Search Progress on CX1-PREF with 16 Orbits

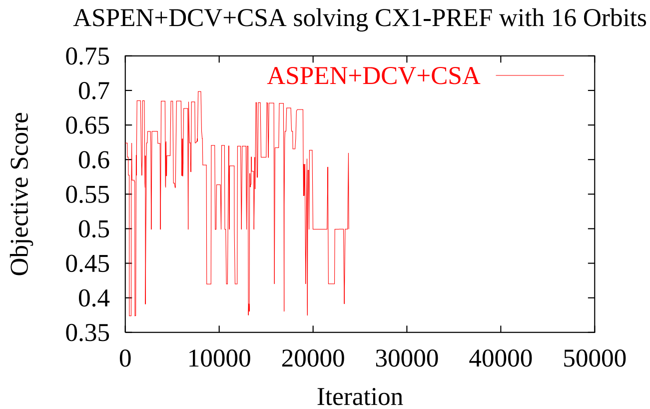
- Conflicts versus Iteration:



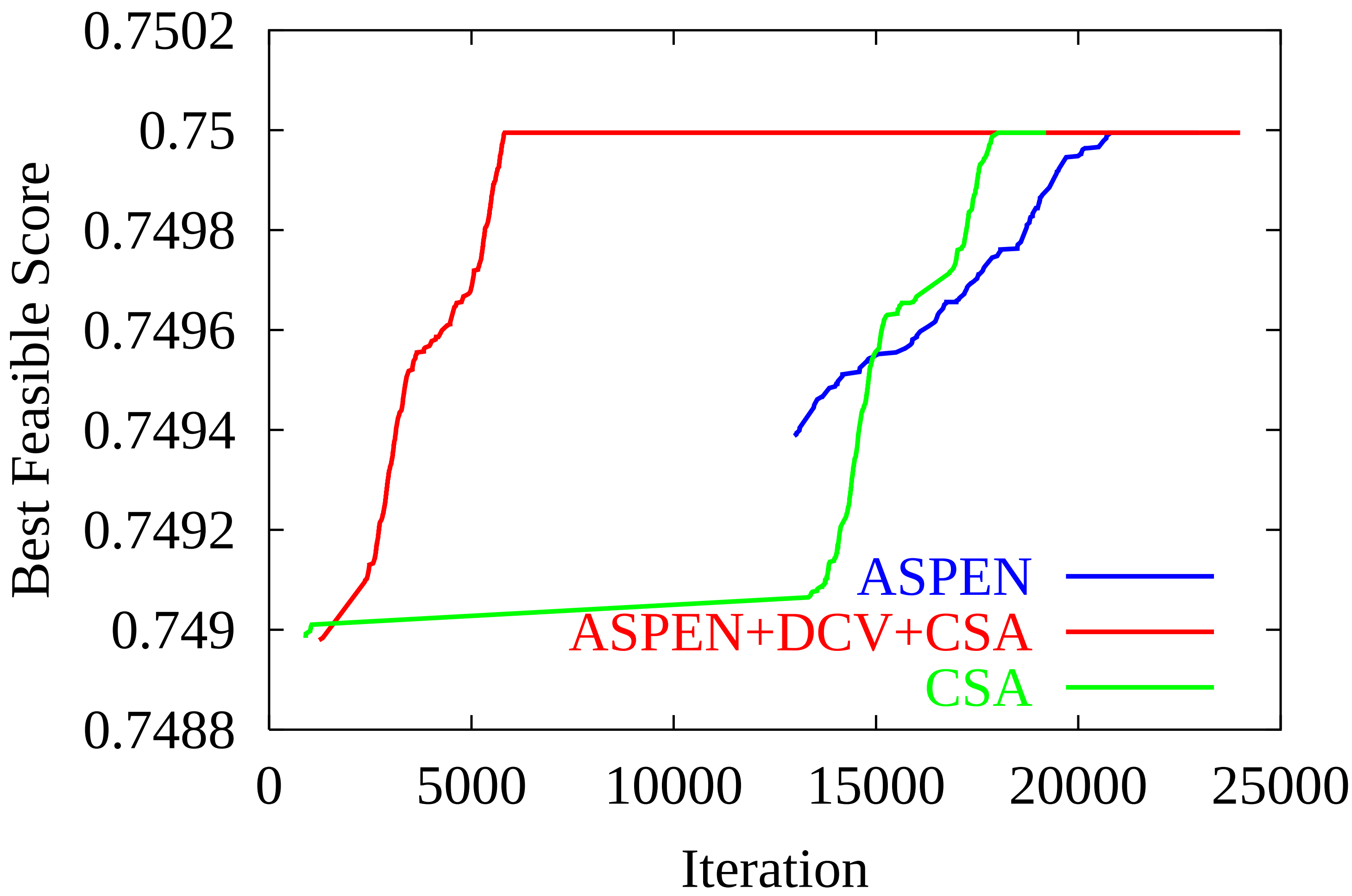


- Score versus Iteration:

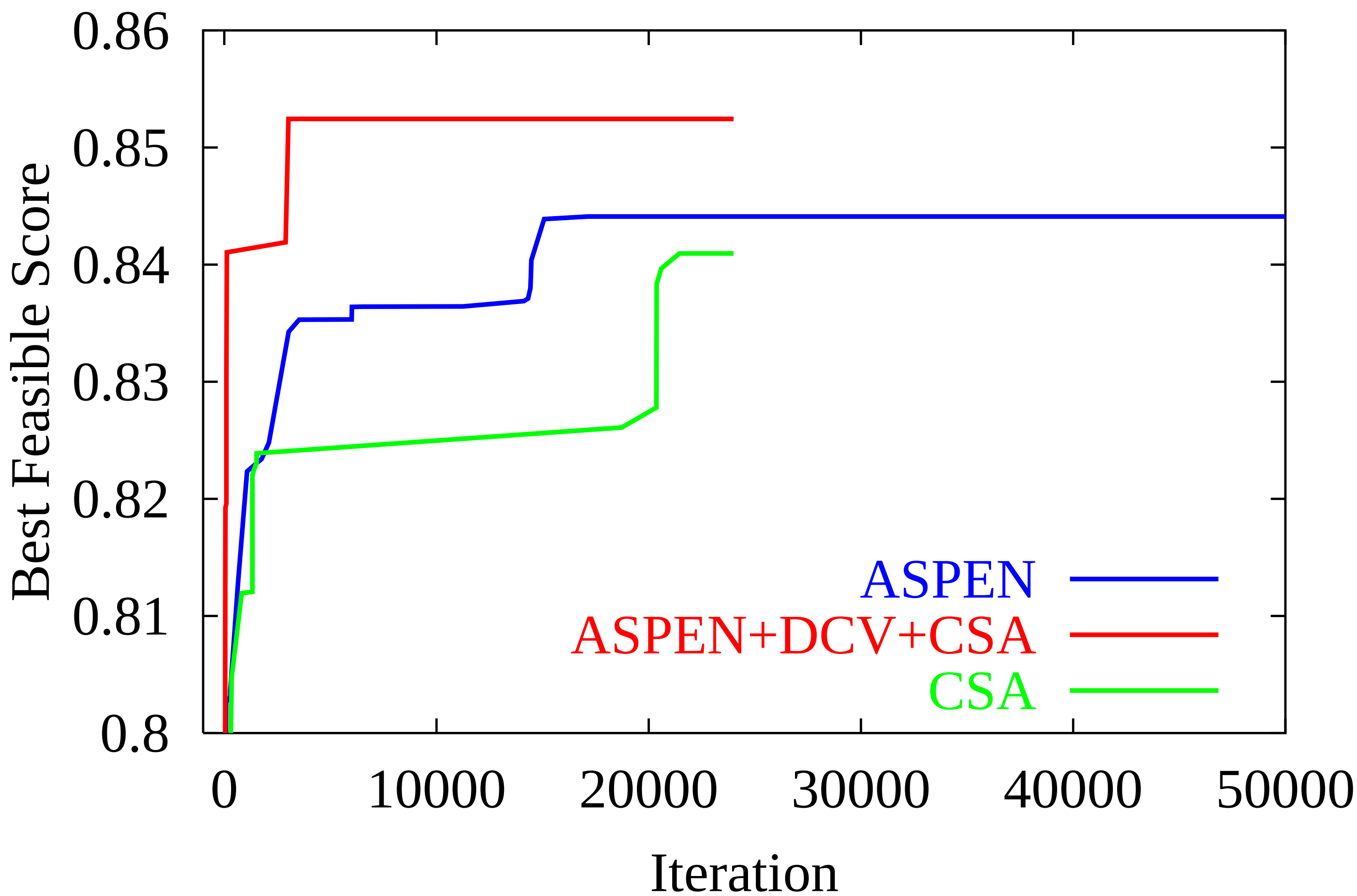




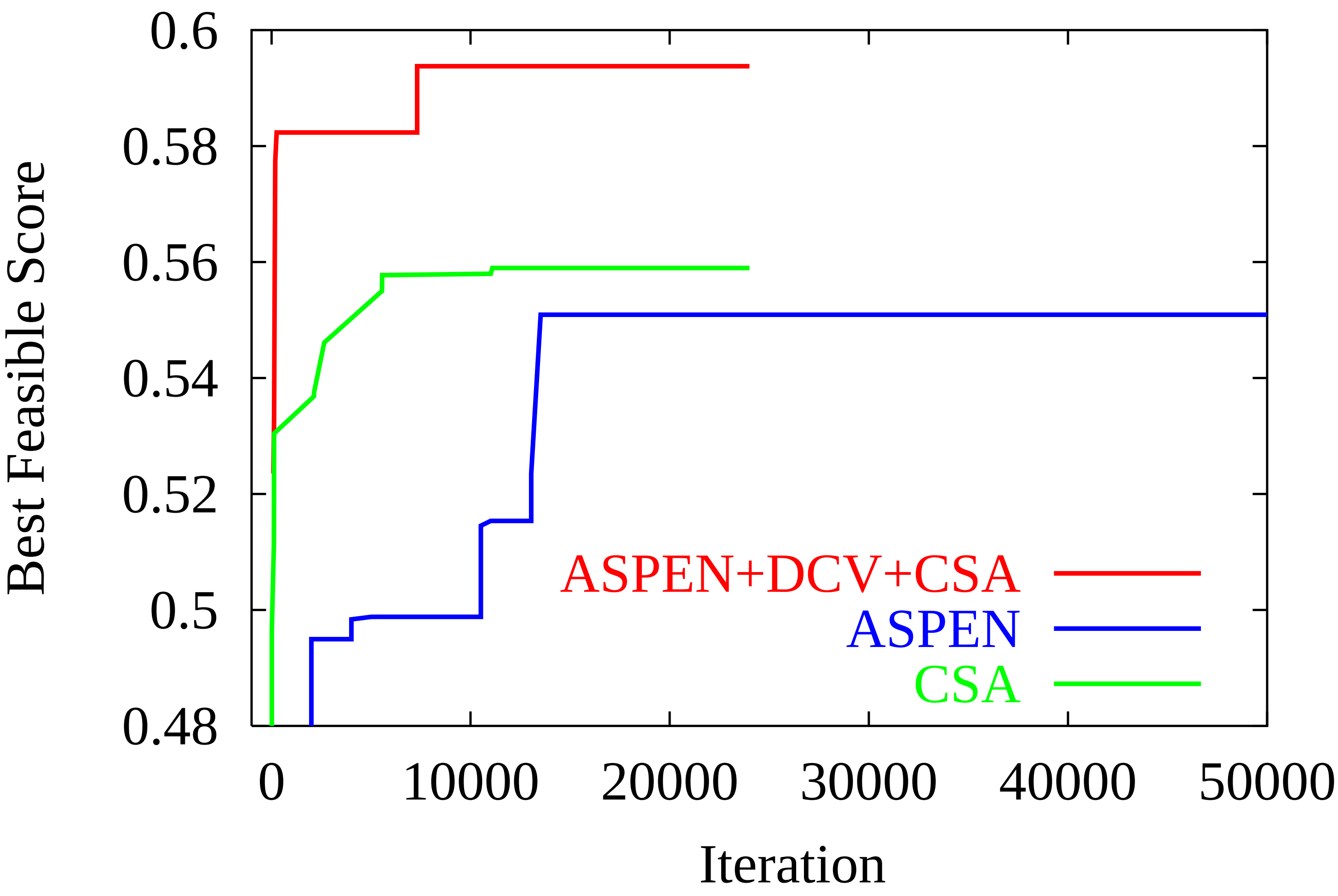
Best Feasible Solution on CX1-PREF with 8 Orbits



Best Feasible Solution on OPTIMIZE Benchmark



Best Feasible Solution on PREF Benchmark



Conclusions

- Partitioning of discrete constrained optimization in temporal planning
  - Systematic method to resolve general constraints across partitions
  - Significant reduction in search space by reducing the base of the exponential complexity
- Few parameters to tune in algorithm
- Extensions to temporal planning in continuous and mixed domains