DISCRETE-SPACE LAGRANGIAN OPTIMIZATION FOR MULTI-OBJECTIVE TEMPORAL PLANNING IN DISCRETE SPACE

Benjamin W. Wah and Yixin Chen
University of Illinois at Urbana-Champaign

Robert Morris
National Aeronautics and Space Administration

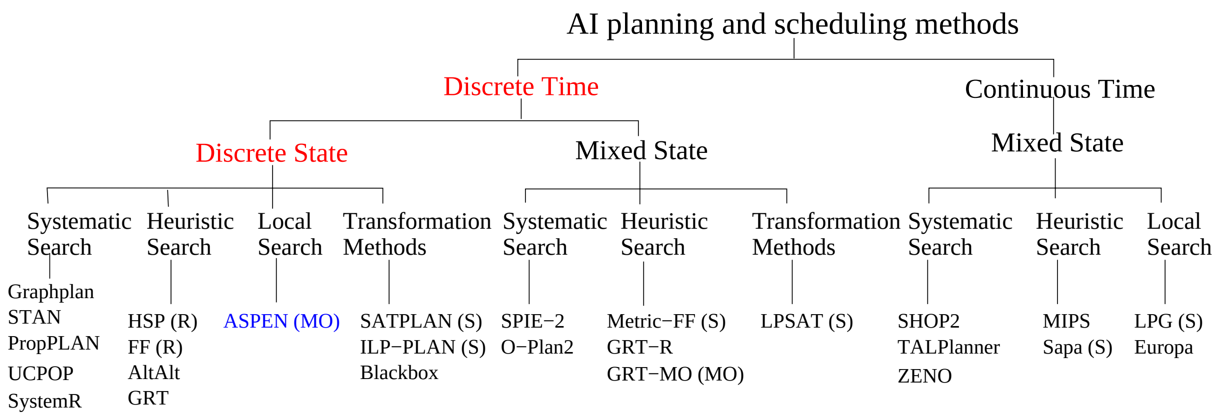
August 9, 2003

Outline

- Introduction
  - Research problem addressed
  - Pareto optimality
- Theory of Lagrange multipliers for discrete constrained optimization
  - Necessary and sufficient extended saddle-point condition
  - Iterative implementation
- Partitioning of variable space
  - Distributed necessary and sufficient extended saddle-point condition
  - Distributed iterative implementation
- Experimental results on ASPEN
- Conclusions

INTRODUCTIONS

A Classification of Existing Approaches in Planning

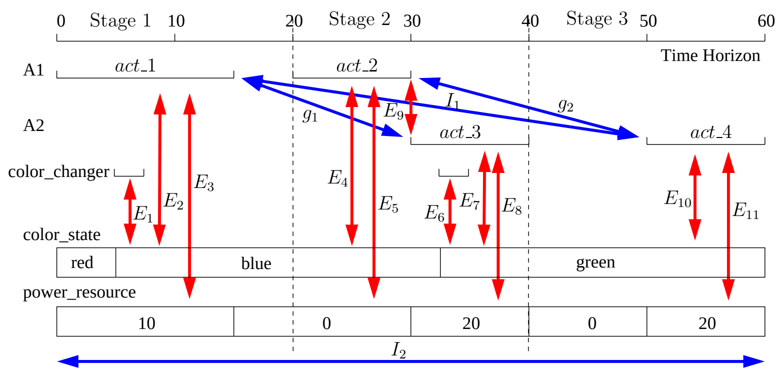


ASPEN

- 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

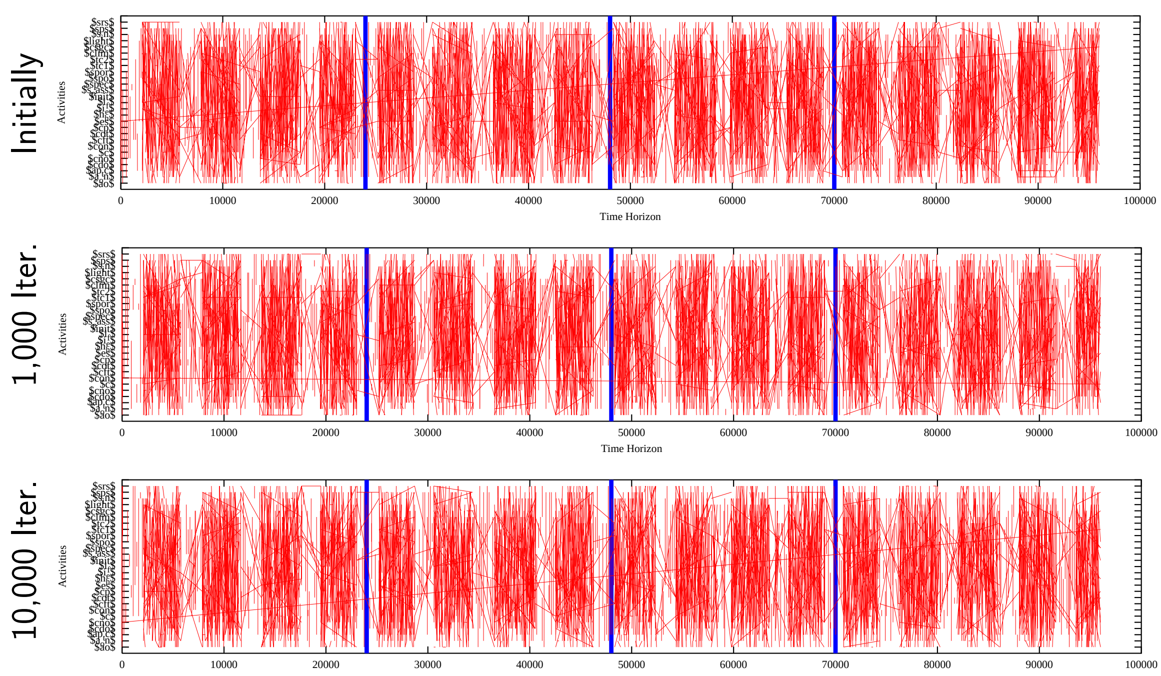
```
model toy {HORIZON_START = 1998-1/00:00:00; horizon_duration = 60s; time_scale = second;};
parameter string color {domain = ("red", "blue", "green");};
State_variable color_sv {states = ("red", "blue", "green"); default_state = "red";};
Resource power {type = non_depletable; capacity = 25; min_value = 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

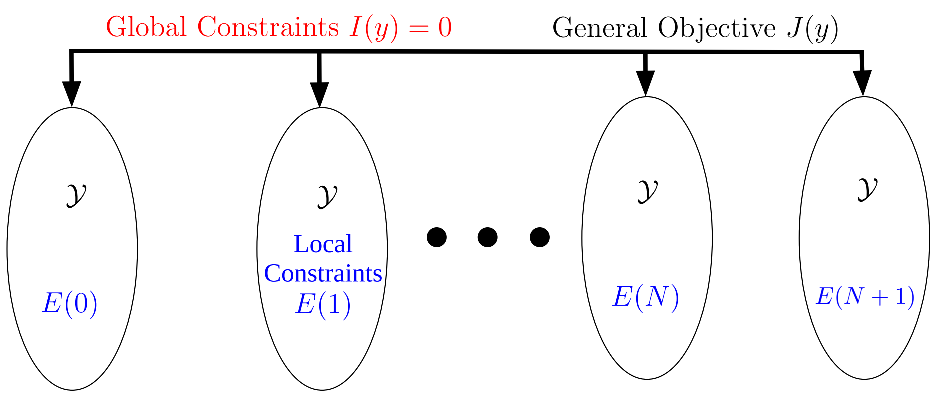


Research Problem Addressed

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

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

Mathematical Formulation



$$\min_y \quad \{J_1(y), J_2(y), \cdots, J_k(y)\}$$

$$\text{subject to} \quad E(j, y(j)) = 0, \quad j = 0, 1, \cdots, N+1$$

$$I(y) = 0$$

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

Multiobjective Optimization and Pareto Optimality

- Optimizing $F(x)$ consisting of a vector of k objective functions:

$$\min_{x} \quad F(x) = (f_1(x), f_2(x), \cdots, f_k(x))^T. \tag{1}$$

- **Pareto optimality**: A *Pareto optimal set* consists of *Pareto optimal solutions* (POS) that are not dominated by any other solutions
  - Solution y *dominates* solution x if x is worse than or equal to y in all objectives, with at least one strictly worse.
- Most search algorithms look for all POS in the Pareto optimal set.

Search for POS

- Weighted-sum method: A new POS can be found by varying the weights and by solving the single-objective problem for each combination of weights.
  - The Pareto optimal set can only be generated when all the objective functions are convex
- Norm method: transforms the multiple objectives into the following single objective with integer p:

$$\min_{x} \left[\sum_{i=1}^{k} w_i \left(\frac{f_i(x) - f_i^*}{f_i^*} \right)^p \right]^{\frac{1}{p}}, \tag{2}$$

  - For finite p, it cannot guarantee that all POS be found, even for all possible combinations of weights.

Counter-Example on the Weighted-Sum Method

- Three solutions: A (16, 40); B (13, 72); C (15, 42)
- Contradiction in finding weights such that C has the minimum weighted sum
  - C is better than A $\Longrightarrow 15W_1 + 42W_2 < 16W_1 + 40W_2 \Longrightarrow 2W_2 < W_1$
  - C is better than B $\Longrightarrow 15W_1 + 42W_2 < 13W_1 + 72W_2 \Longrightarrow W_1 < 15W_2$
- Scenario happens because the second objective function is not convex

Search for POS: Minimax Method

- Minimax method: a special case of (2) when $p = \infty$ and $f_i^* = 0$

$$\min_x \left\{ \max_{i=1}^{k} \left[w_i f_i(x) \right] \right\}. \tag{3}$$

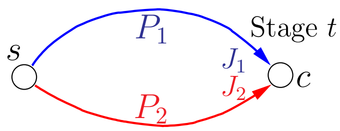
  potentially generate all POS for nonconvex problems

- Use the minimax approach to formulate a multi-objective planning problem as a *single-objective dynamic optimization problem with equality constraints as follows:*

$$\begin{aligned} \min_y \quad & J[y] = \max_{i=1}^{k} \left[w_i J_i(y) \right] \\ \textit{such that} \quad & E(t, y(t)) = 0, \qquad t = 0, \cdots, N+1 \\ \textit{and} \quad & I[y] = 0, \end{aligned} \tag{4}$$

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
$J_2 \leq J_1 \implies P_2 \rightarrow P_1$

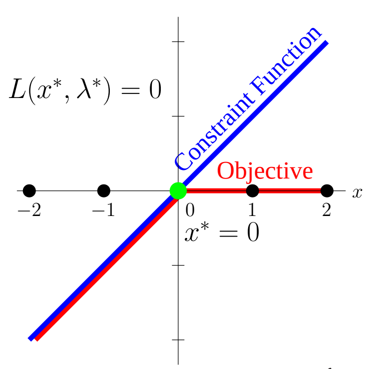
  - Polynomial worst-case complexity: $O\left(N|\mathcal{Y}|^2\right)$
- Path dominance is not applicable when there are global constraints
  - A dominating path early on may become infeasible due to global constraints that got violated later
  - Exponential search space: $O\left(|\mathcal{Y}|^{N+2}\right)$

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

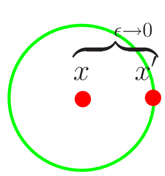
$$L(-1, \lambda^*) \geq L(0, \lambda^*) \leq L(1, \lambda^*)$$

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

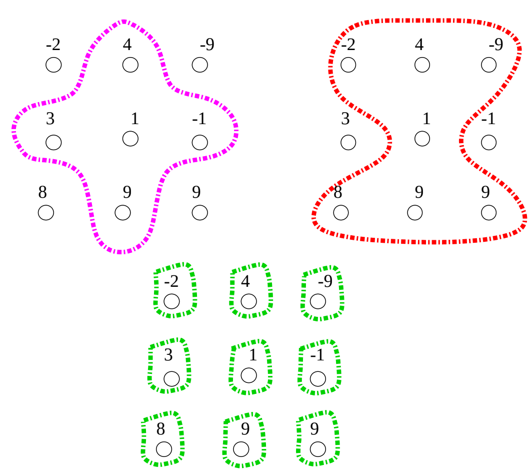
THEORY OF EXTENDED SADDLE POINTS FOR DISCRETE CONSTRAINED OPTIMIZATION

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



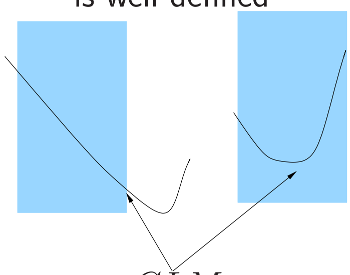
Discrete Space: $\mathcal{N}_{dn}(x)$
x is a vector of discrete variables
User defined neighborhood



Constrained Local Minimum (CLM)

Continuous Space: CLM_{cn}

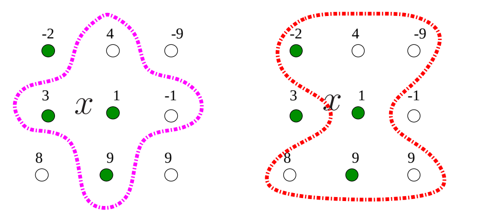
- Feasible local minimum when compared to feasible points inside an open sphere
- Whether point x is a CLM_{cn} is well defined



CLM_{cn}

Discrete Space: CLM_{dn}

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



CLM_{dn} at x No CLM_{dn} at 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 $x = 0$

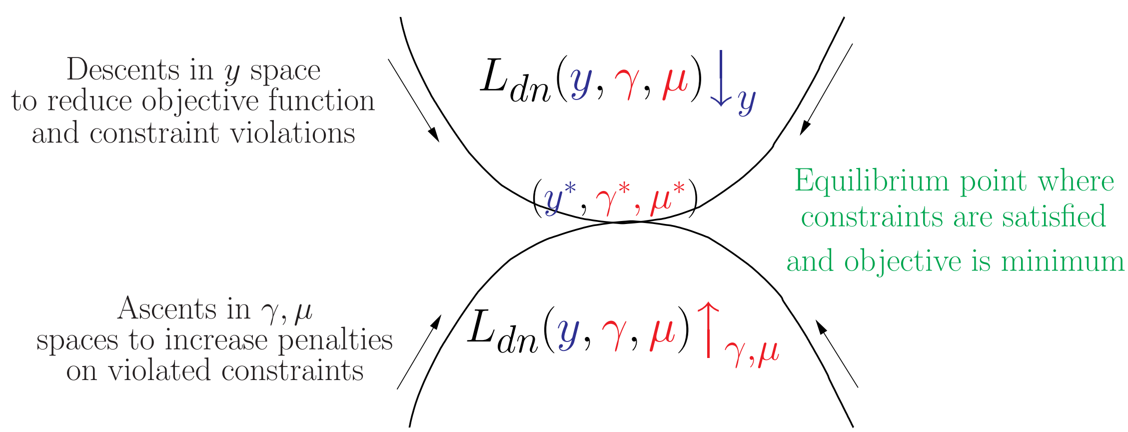
$$L_{dn}(y, \gamma, \mu) = J(y) + \sum_{t=0}^{N+1} \gamma^T(t) \cdot \Big|E(t, y(t)\Big| + \mu^T \cdot \Big|I(y)\Big|$$

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

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

- (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 $\gamma^{**} > \gamma^*$ and $\mu^{**} > \mu^*$

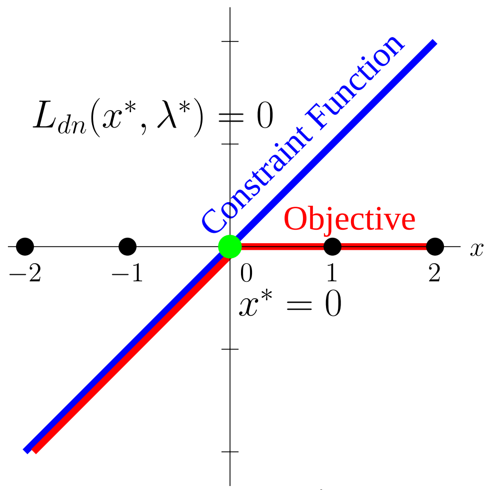
Intuitive Meaning Behind Saddle Points



Although γ^* and μ^* always exists,

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

$$L_{dn}(-1, \lambda^*) > L_{dn}(0, \lambda^*) < L_{dn}(1, \lambda^*)$$

leads to $\lambda^* > 1$

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

Iterative Implementation

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

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

Outer Loop

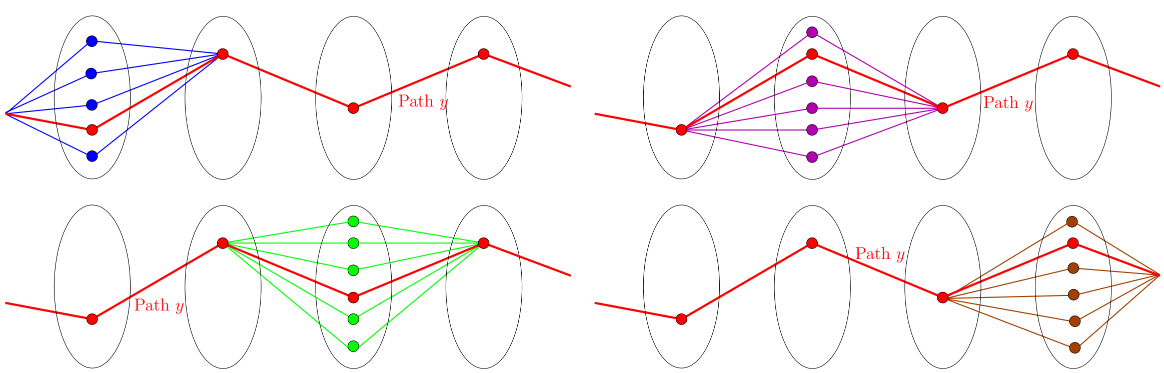
$L_{dn}(y, \gamma, \mu)\downarrow_{y}$ to find y^*

Inner Loop

PARTITIONING OF VARIABLE SPACE

Partitionable Neighborhoods

$\mathcal{N}_p(y)$ (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 $\mathcal{N}_p(y)$ 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 & Sufficient ESPC for CLM_{dn}

- Path y is a CLM_{dn} if and only if it satisfies
  - Distributed Necessary & Sufficient ESPC for all $t = 0, 1, \cdots, N+1$

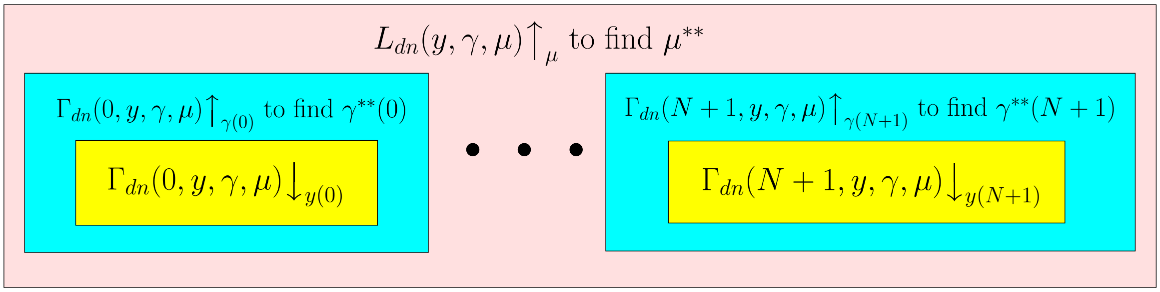
    $$\Gamma_{dn}(t, y^*, \gamma(t)', \mu^*) \le \Gamma_{dn}(t, y^*, \gamma(t)^*, \mu^*) \le \Gamma_{dn}(t, y', \gamma(t)^*, \mu^*)$$

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

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

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

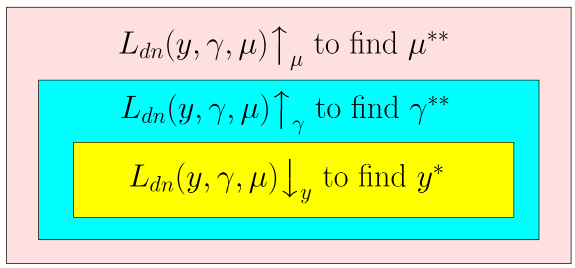
Iterative Implementation



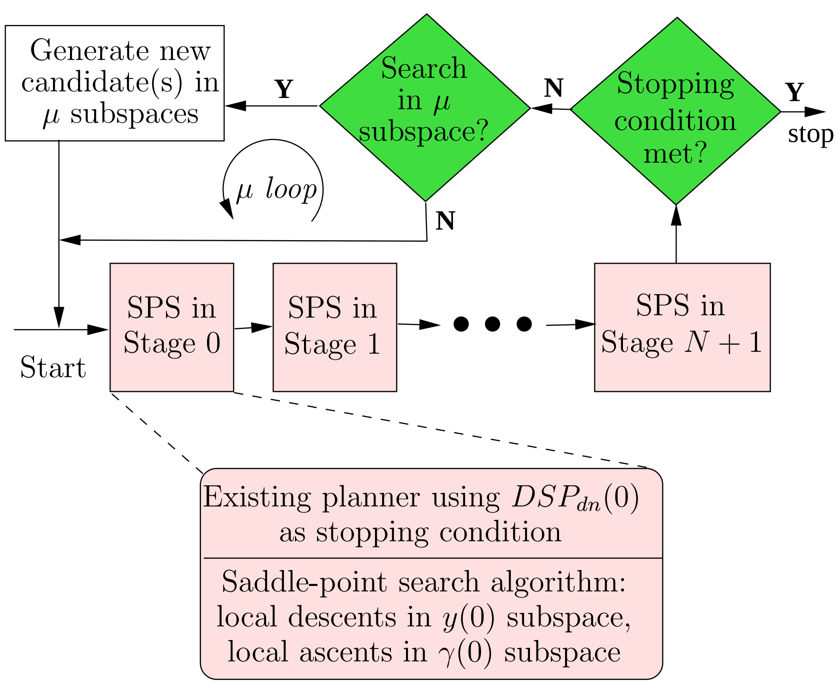
Observation:

- Based on the partitionable neighborhood defined, the combined local minimum of Γ_{dn} in all subspaces is the local minimum of L_{dn}

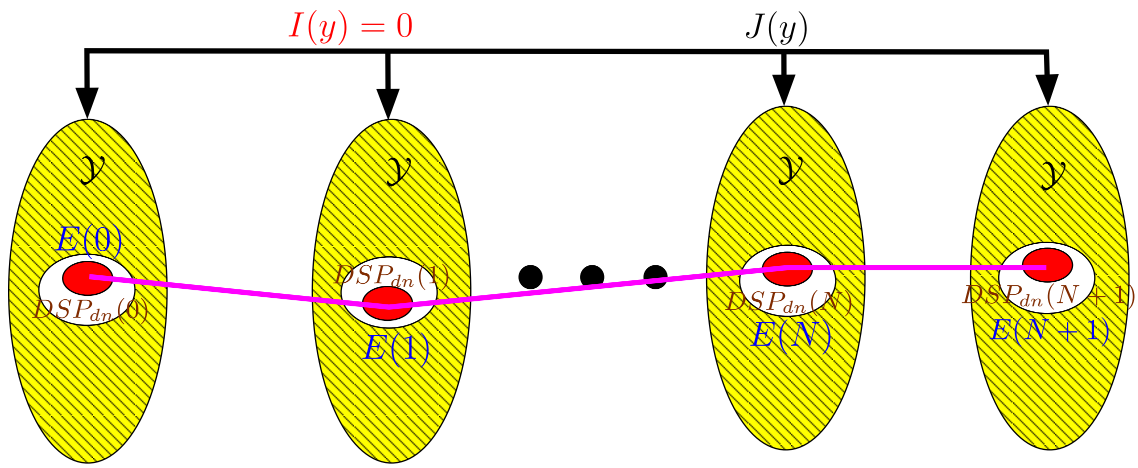
Equivalent Search:



Heuristic Search Procedure for Finding Local Optimal Paths



Reduced Search Space for Finding Feasible/Optimal Paths



Significant reduction in complexity

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 (using existing planner to solve partitioned subproblem)
   - Evaluate augmented distributed Lagrangian function of stage t:

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

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

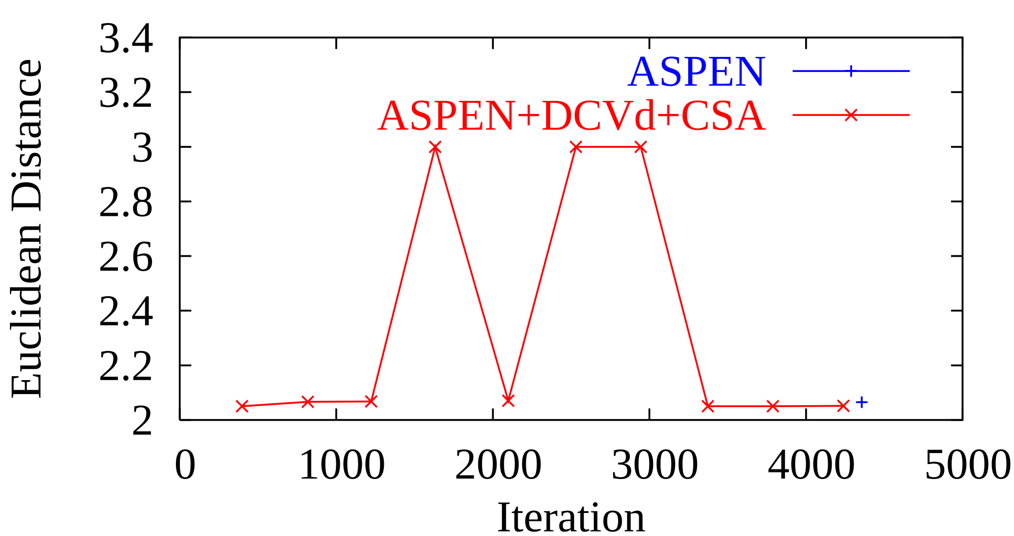
    $perl \quad probgen.pl \quad < random\ seed > \quad < number\ of\ orbits >$

- In default ASPEN search, repeat the following steps

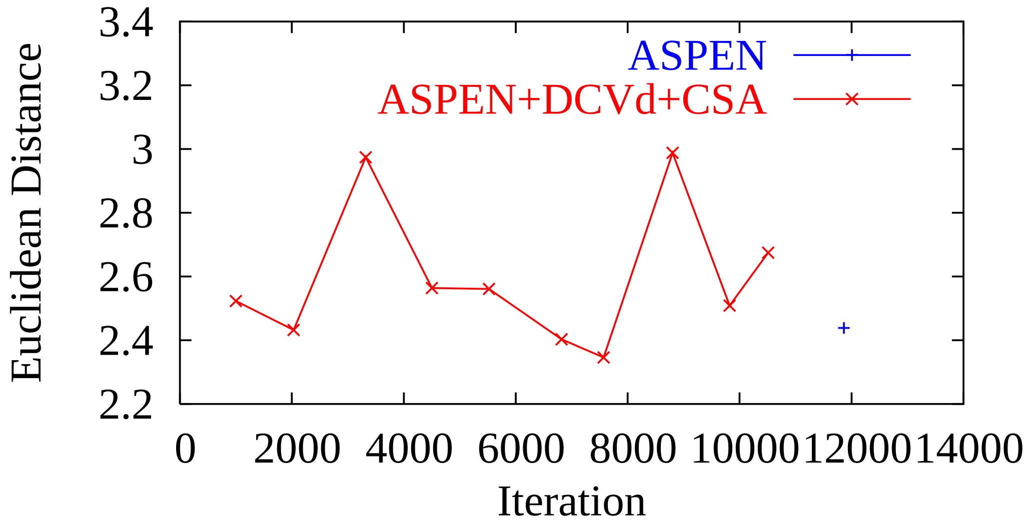
  a) Find feasible schedule using *repair*

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

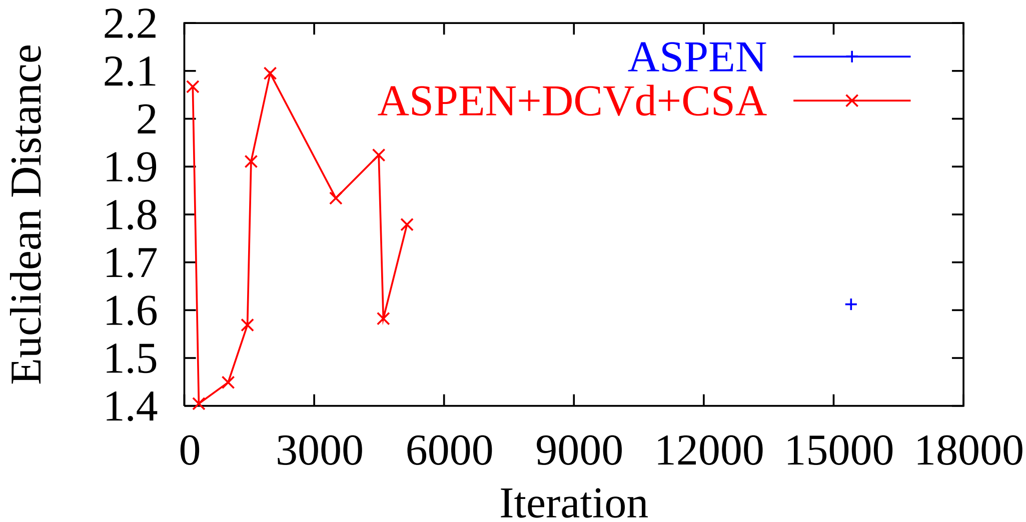
Pareto Solutions on CX1-PREF Benchmark with 8 Orbits



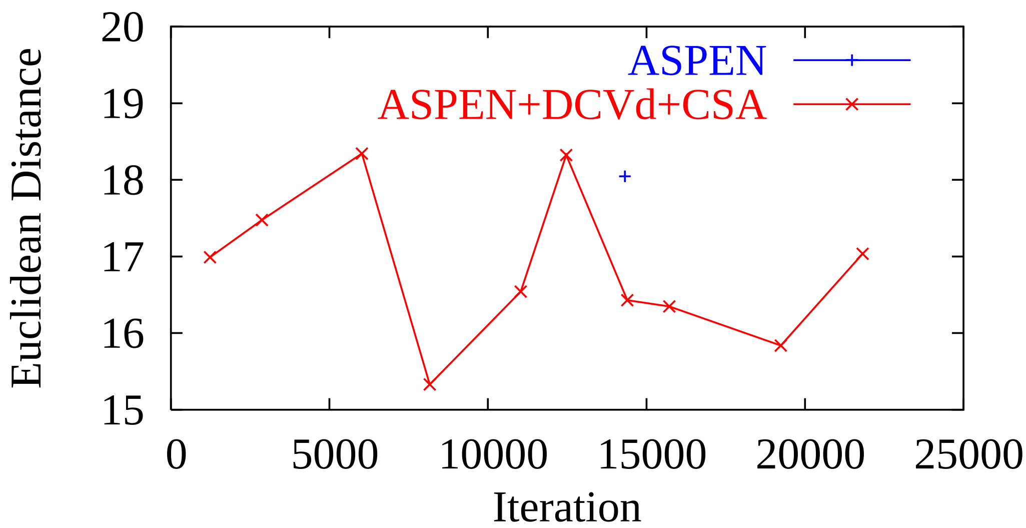
Pareto Solutions on CX1-PREF Benchmark with 16 Orbits



Pareto Solutions on OPTIMIZE Benchmark



Pareto Solutions on PREF Benchmark



Conclusions

- Partitioning of discrete constrained optimization in temporal planning
  - Distributed method to resolve global 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