להלן סיכום של המידע מהמקורות שסיפקת, המתמקד במושגי יסוד וטכניקות חיפוש ופתרון לבעיות אופטימיזציה בדידה:

שתי ההרצאות עוסקות במטא-היוריסטיקות וטכניקות לפתרון בעיות אופטימיזציה, בדגש על בעיות NP-שלמות. מושג בסיסי הוא **שכונה** ,**(Neighborhood)** המוגדרת כקבוצת הפתרונות הנגישים מפתרון נוכחי באמצעות שינוי קטן.

מטא-היוריסטיקות הן שיטות ברמה גבוהה יותר שמטרתן למצוא או ליצור היוריסטיקות טובות לפתרון בעיות אופטימיזציה. מטרתן העיקרית היא **לברוח מאופטימה מקומית** ולשאוף לאופטימה גלובלית, אם כי הן אינן מבטיחות מציאת האופטימה הגלובלית. הן פועלות על ידי הנחיית היוריסטיקות ברמה נמוכה יותר (כמו חיפוש מקומי) והתאמת אופרטורי חיפוש או פרמטרים לבעיות אופטימיזציה חדשות. דוגמאות למטא-היוריסטיקות כוללות חיפוש מקומי איטרטיבי ,(ILS) חיפוש מבוסס טאבו ,(Tabu Search) חישול ממוחשב ,(Simulated Annealing) אופטימיזציית להקת נמלים (ACO) וחיפוש שכנות גדולות אדפטיבי.(ALNS)

היוריסטיקות רב-שלביות (Multi-Stage Heuristics) הן טכניקה שמטרתה למנוע סריקה של כל השכונה על ידי פיצול תהליך הבחירה לשלבים. לדוגמה, בבעיית ,N-Queens ניתן לבחור בשלב הראשון את המשתנה עם הכי הרבה הפרות, ובשלב השני את הערך עם הכי פחות הפרות כתוצאה מהבחירה. דוגמאות נוספות כוללות סודוקו (בחירת תא ריק ואז הקצאת ערך) ובעיית הסוכן הנוסע (TSP)באמצעות הכנסה חמדנית (בחירת עיר התחלה ואז בניית מסלול על ידי הוספת העיר המגדילה הכי פחות את אורך המסלול).

חיפוש מקומי איטרטיבי (ILS) הוא שינוי של חיפוש מקומי או טיפוס גבעה שמטרתו לשפר את איכות הפתרונות בבעיות אופטימיזציה בדידות. המנגנון המרכזי שלו הוא **חיפושים מקומיים חוזרים ונשנים** מתצורות התחלתיות מגוונות, בשילוב עם **פרטורבציה** לבריחה מאופטימה מקומית. המטרה היא לאזן בין **חקירה (exploration)** של מרחב החיפוש לבין **ניצול (exploitation)** של פתרונות טובים שנמצאו. המבנה הכללי של אלגוריתם ILS כולל יצירת פתרון התחלתי, ביצוע חיפוש מקומי, שמירת הפתרון הטוב ביותר שנמצא עד כה, ויצירת פתרון חדש (באמצעות פרטורבציה) להמשך האיטרציה.

חישול ממוחשב (Simulated Annealing - SA) הוא מטא-היוריסטיקה בהשראת פיזיקה סטטיסטית. הוא מקבל מהלכים המשפרים את פונקציית המטרה, אך גם **מקבל מהלכים גרועים יותר באופן הסתברותי** כדי למנוע היתקעות. ההסתברות לקבלת מהלך גרוע תלויה בכמה המהלך גרוע ובכמה האלגוריתם התקדם בחיפוש) באמצעות פרמטר "טמפרטורה .(T "ה**"לוח זמנים של חישול"** קובע כיצד הטמפרטורה משתנה לאורך זמן. מתחילים עם טמפרטורה גבוהה (המקבילה להליכה אקראית), ומקררים אותה ככל שהחיפוש מתקדם, מה שהופך את החיפוש לחמדני יותר; ב-T=0, החיפוש הופך לטיפוס גבעה. הכלל המטרופוליס קובע את הסתברות הקבלה SA .יכול לשמש כבסיס ל.ILS-

חיפוש מבוסס טאבו (Tabu Search) הוא מטא-היוריסטיקה העובדת עם מצב יחיד, בדומה לטיפוס גבעה, אך מאפשרת לעבור לפתרון השכן הטוב ביותר **אפילו אם הוא גרוע יותר** (ירידה במדרון). כדי להימנע מלכידה באופטימה מקומית או מחזורים, הוא **משתמש בזיכרון (רשימת טאבו)** כדי להימנע מלחזור למצבים שביקר בהם לאחרונה או מלבצע אופרטורים שבוצעו לאחרונה. רשימת הטאבו מתחזקת את ה"שכונה העוינת" של מצבים שאסור לבקר בהם כרגע. ה**"טאבו טנור Tabu) " Tenure)**הוא פרמטר הקובע כמה זמן מצב או אופרטור נשארים ברשימת הטאבו. במקום לשמור

מצבים שלמים (שעשוי להיות יקר), ניתן לשמור אופרטורים או תכונות של הבעיה ברשימת הטאבו . **קריטריון שאיפה (Aspiration Criteria)** מאפשר לעקוף חוק טאבו אם המהלך האסור מוביל לפתרון טוב משמעותית מהפתרון הטוב ביותר שנמצא עד כה. טכניקות נוספות בחיפוש טאבו כוללות **התעצמות (Intensification)** על ידי חזרה לפתרונות איכותיים שנמצאו בעבר ו**גיוון (Diversification)**על ידי הכנסת אקראיות לבריחה ממלכודות מקומיות .**תנודה אסטרטגית (Strategic Oscillation)**היא רעיון של תנועה מכוונת בין אזורים ברי-ביצוע (feasible) ולא ברי-ביצוע (infeasible) במרחב החיפוש כדי לחקור יותר ולברוח מאופטימה מקומית.

אופטימיזציית להקת נמלים (ACO) היא מטא-היוריסטיקה בהשראת התנהגות נמלים המחפשות מזון. היא מנצלת מנגנון משוב חיובי (באמצעות פרומונים) ומבנה נתונים גלובלי (שבילי פרומון) המשתנה באופן דינמי ככל שנמלים חוצות מסלולים. לכל נמלה יש הסתברות לעבור בין מצבים התלויה בכמות הפרומון על השביל ובהיוריסטיקה מקומית (נראות השביל). עדכון הפרומון כולל **אידוי (evaporation)** (החלשת שבילים ישנים) ו**הפקדה (deposit)** (הוספת פרומון לשבילים בהם עברו נמלים, לרוב בהתאם לאיכות הפתרון שנמצא). קצב האידוי (ρ) מאזן בין חקירה ρ) גבוה (לניצול ρ)נמוך ACO .(יכולה להיתקע באופטימה מקומית, וטכניקות להתמודדות כוללות הפעלה מחדש, הגברת החקירה, שינוי סכימת העדכון ושימוש בטכניקות גיוון נוספות כמו ריבוי קולוניות או הכללת זיכרון של נמלים. בגרסה מסוימת ACO ,עשויה להתנהג כמו ILS על ידי התחלה מדגימות אקראיות ובניית פתרונות בהנחיה, תוך שימוש בפרומונים לניצול ובלמידת הפתרון הטוב ביותר עד כה.

חיפוש שכנות גדולות אדפטיבי (Adaptive Large Neighborhood Search - ALNS) היא מטא-היוריסטיקה יעילה במיוחד לפתרון בעיות כמו ניתוב כלי רכב, תזמון ובעיות אריזה/ניתוב שונות . ALNSמשפרת פתרון התחלתי באופן איטרטיבי על ידי חקר שכונה שלו באמצעות **אופרטורים של "השמדה" ו"תיקון (destroy/repair pairs)** .ליבת הגישה היא **אדפטיבית** ,כלומר, היא **בוחרת באופן דינמי אופרטור שכונה** מתוך מאגר על בסיס התפלגות הסתברותית המתעדכנת במהלך החיפוש בהתאם להצלחת כל אופרטור בשיפור איכות הפתרון. החלטה אם לקבל או לדחות פתרון חדש מבוססת על קריטריון קבלה מוגדר מראש, למשל, כלל הקבלה של חישול ממוחשב (Metropolis step). התהליך נמשך עד שעומד תנאי עצירה כמו הגבלת זמן או מספר איטרציות.

ההרצאות גם דנות ביחס בין **היתכנות (Feasibility)** ל**אופטימליות (Optimality)** בבעיות אופטימיזציה בדידה.

- **היתכנות** פירושה שהפתרון עומד בכל אילוצי הבעיה. בחיפוש מקומי, ניתן להתמקד בחקירת שכונה של פתרונות ברי-ביצוע בלבד.
- **אופטימליות** קשורה למקסום או מזעור פונקציית מטרה .מתן אפשרות ל**פתרונות לא ברי-ביצוע (infeasible)** באופן זמני עשוי לאפשר חקירה רחבה יותר של מרחב החיפוש ובריחה מאופטימה מקומית, מתוך כוונה לתקן את ההפרות בהמשך. בשיטות כמו אלגוריתמים גנטיים, ניתן לאפשר פתרונות לא ברי-ביצוע ולהעניש אותם באמצעות פונקציית ההתאמה (fitness).

בעיות אופטימיזציה בדידה-NP שלמות שמוזכרות ונדונות כוללות:

- **אריזת בינים :(Bin Packing)** מטרה למזער את מספר הבינים בקיבולת קבועה הנדרשים לארוז סט פריטים בגדלים שונים.

- **בעיית התיק :(Knapsack Problem)** בהינתן סט פריטים עם משקל וערך, לבחור תת-קבוצה של פריטים למקסימום ערך כולל, תחת אילוץ משקל מקסימלי לתיק. בעיית **0-1 Knapsack**היא גרסה בה כל פריט נבחר בשלמותו או לא נבחר כלל. ניתן לפתור אותה באמצעות **תכנון דינמי**)גישת Bottom-up לפתרון תת-בעיות (או **Branch and Bound**)פיצול הבעיה לתת-בעיות והערכת חסמים תוך שימוש ב ,Relaxationכמו Fractional Knapsack).אסטרטגיות חיפוש ב BB-כוללות DFS, Best First Search ו Least-Discrepancy Search (LDS).

- **בעיית התיק המרובה :(Multiple Knapsack Problem - MKP)** הרחבה של בעיית התיק עם מספר תיקים, לכל אחד מגבלת משקל משלו, במטרה למקסם את הערך הכולל של פריטים שמוקצים לתיקים מבלי לחרוג מקיבולת כל תיק. כל פריט נכנס לתיק אחד בלבד או לא נכנס כלל.

- **צביעת גרפים :(Graph Coloring)** מציאת המספר המינימלי של צבעים הנדרשים לצביעת קודקודי גרף כך שאין שני קודקודים שכנים בעלי אותו צבע. מציאת המספר הכרומטי של גרף או קביעת האם גרף הוא-k צביע הן בעיות-NP שלמות. נדונות שלוש גישות לחיפוש מקומי עבור בעיה זו :

1. התמקדות בהיתכנות בלבד, על ידי המרה לרצף של בעיות היתכנות) למשל, התחלה בצביעה חוקית עם k צבעים וניסיון להגיע לצביעה חוקית עם k-1 צבעים תוך תיקון הפרות .(ניתן לשלב זאת עם אלגוריתמים גנטיים לתיקון.

2. התמקדות באופטימליות בלבד, תוך הנחה שהפתרון בר-ביצוע, ושאיפה למזער את מספר הצבעים בעקיפין (למשל, על ידי מקסום סכום גודלי מחלקות הצבע בריבוע). ניתן להשתמש ב"שרשראות קמפה (Kempe Chains) "כאופרטור שכונה השומר על היתכנות מקומית.

3. גישה היברידית, המשלבת מזעור הפרות (אי-היתכנות) עם אופטימיזציה של פונקציית המטרה (מספר הצבעים) בפונקציית מטרה משוקללת או מאוחדת. הוצגה פונקציית מטרה ספציפית המשלבת מזעור "קצוות רעים" (קצוות המחברים קודקודים באותו צבע) עם מקסום גודל מחלקות הצבע, המבטיחה שמינימום מקומי הוא בהכרח צביעה חוקית (בראיה).

   - אלגוריתם ICN (Iterative Coloring Neighbourhood Search) הוא גישה נוספת לצביעת גרפים המתמקדת בהעברת קודקודים ממחלקת "אימפאס" (לא צבועים) למחלקות צבע קיימות תוך שמירה על היתכנות המחלקה, ושאיפה לרוקן את מחלקת האימפאס. הוא משתמש בפונקציית התאמה הממזערת את דרגת הקודקודים הלא צבועים ומחיל כללי טאבו ושאיפה.

- **בעיית הסוכן הנוסע :(Traveling Salesperson Problem - TSP)** מציאת המסלול הקצר ביותר העובר דרך סט ערים בדיוק פעם אחת וחוזר לנקודת ההתחלה.

- **בעיית ניתוב כלי רכב בעלי קיבולת - (Capacitated Vehicle Routing Problem CVRP):**בהינתן מחסן, סט לקוחות עם דרישות, וצי רכבים בעלי קיבולת קבועה, מציאת הקצאת מסלולים אופטימלית לרכבים (יוצאים מהמחסן, מבקרים כל לקוח פעם אחת, חוזרים

למחסן, מספקים את כל הדרישות) כדי למזער את אורך המסלול הכולל. זו בעיית-NP שלמה. ניתן לפתור אותה בגישות שונות :

- **היוריסטיקות קונסטרוקציה** (בונות פתרון צעד אחר צעד), כמו היוריסטיקות הכנסה (Inserting Heuristics)או אלגוריתם החיסכון של קלארק ורייט (Clarke and Wright Savings Algorithm).
- **היוריסטיקות שיפור איטרטיבי**)משפרות פתרון קיים.(
- **גישה דו-שלבית**)למשל, שימוש ב MKP-להקצאת לקוחות לרכבים על בסיס דרישה וקיבולת, ואז פתרון TSP עבור כל רכב.(
- **מטא-היוריסטיקות** כמו ILS, GA, Tabu Search, SA, ACO, ו ALNS. ALNS-יעיל במיוחד עבור CVRP, תוך שימוש באופרטורים כמו העברה (Relocate), החלפה (Swap), החלפה צולבת (Cross-Exchange) ו.Two-Opt-

- **בעיית טורניר הנוסעים :(Traveling Tournament Problem - TTP)** בעיית תזמון מורכבת הכוללת קבוצות המשחקות זו מול זו לאורך עונה, עם אילוצי היתכנות מורכבים (כמו מספר משחקי בית/חוץ רצופים) ומטרת מזעור מרחק נסיעה כולל.

בנוגע לפתרון בעיות אופטימיזציה בדידה, חשוב לזכור שייתכנו פתרונות מרובים, יש צורך בפונקציית מטרה/היוריסטיקה יעילה מבחינה חישובית, ונדרשות שיטות להתמודדות עם אופטימה מקומית) כמו ILS, ובקרת מוטציה, ניצ'ינג .(מומלץ להתחיל עם אלגוריתם חמדני בסיסי (למשל, השכן הקרוב ביותר).